Fig. 1

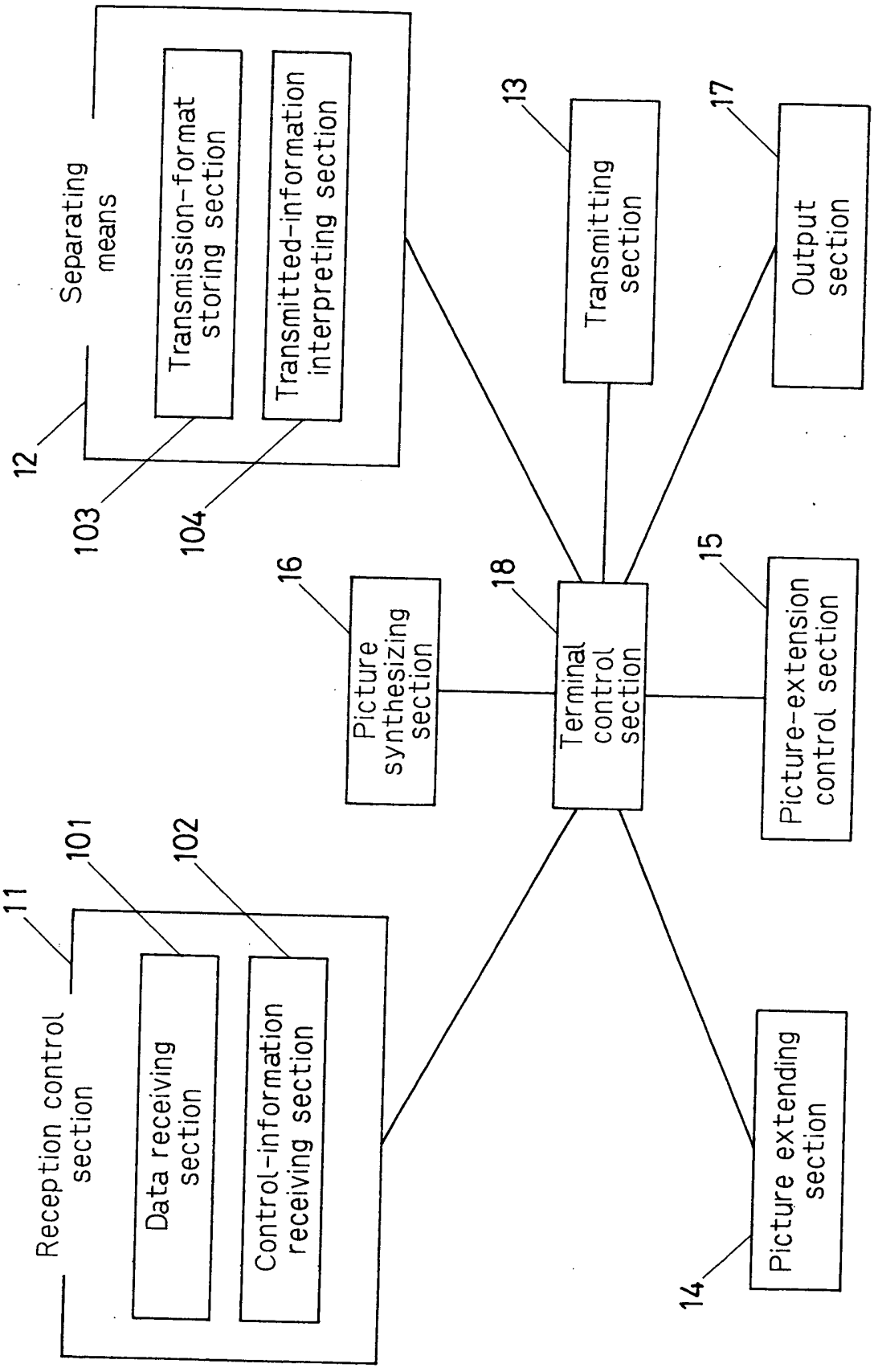

Reception control section
11
Data receiving section
101
Control-information receiving section
102
Separating means
12
Transmission-format storing section
103
Transmitted-information interpreting section
104
Transmitting section
13
Picture extending section
14
Picture-extension control section
15
Picture synthesizing section
16
Terminal control section
18
Output section
17

Fig. 2

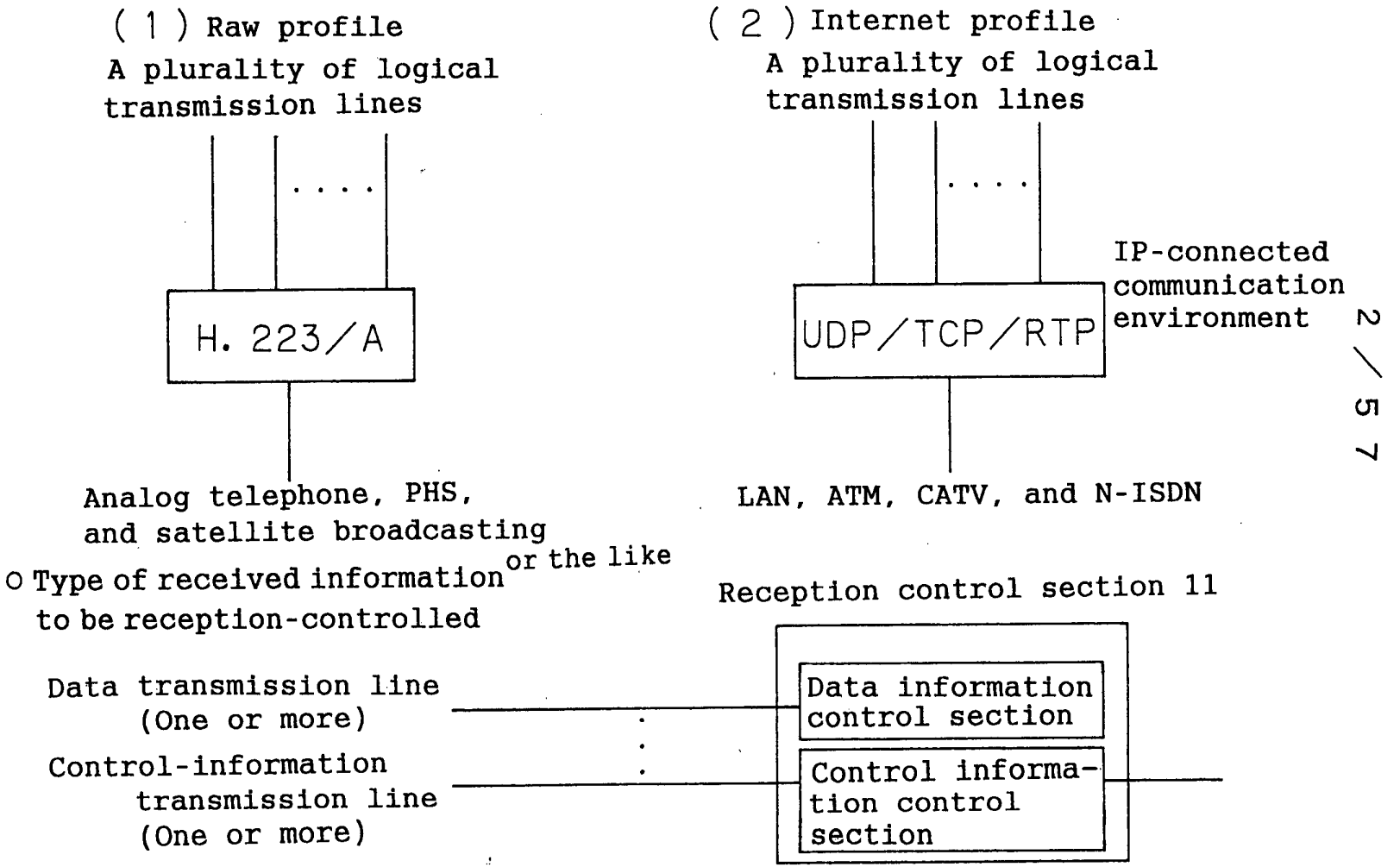

(1) Raw profile
A plurality of logical transmission lines
H. 223/A
Analog telephone, PHS, and satellite broadcasting or the like
(2) Internet profile
A plurality of logical transmission lines
UDP/TCP/RTP
IP-connected communication environment
LAN, ATM, CATV, and N-ISDN
o Type of received information to be reception-controlled
Reception control section 11
Data transmission line (One or more)
Control-information transmission line (One or more)
Data information control section
Control information control section

Fig. 3

○ Information showing start position capable of processing data or not

- Flag for random access (Random access flag), e.g. Intra-frame (I-picture) in the case of picture
- Flag showing access unit (Access flag), e.g. Frame in the case of picture, GOB unit

AL : Adaptation layer
ES : Elementary stream
PTS : Presentation·time·stamp

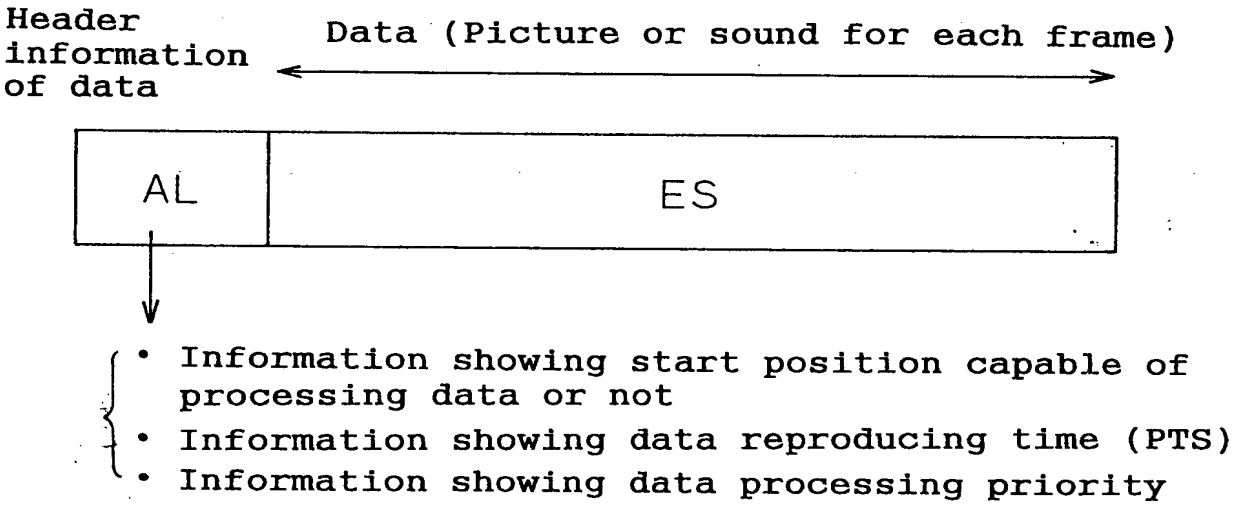
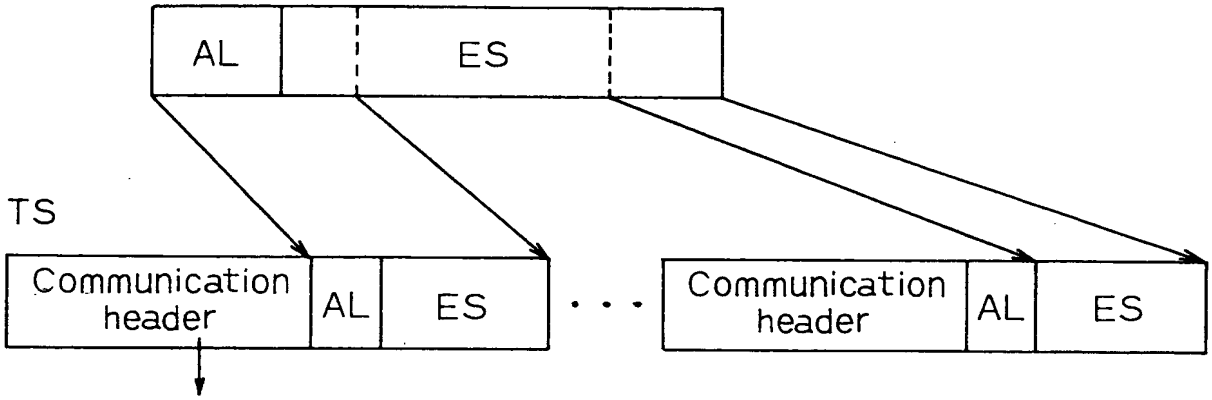


Fig. 4

○TS:Transport stream(Transmission packet)



- Information showing start position capable of processing pieces of data or not
- Identification number for showing data sequence(Sequence number)
- Time concerned with transmission of pieces of data

○Handling time stamp and marker bit

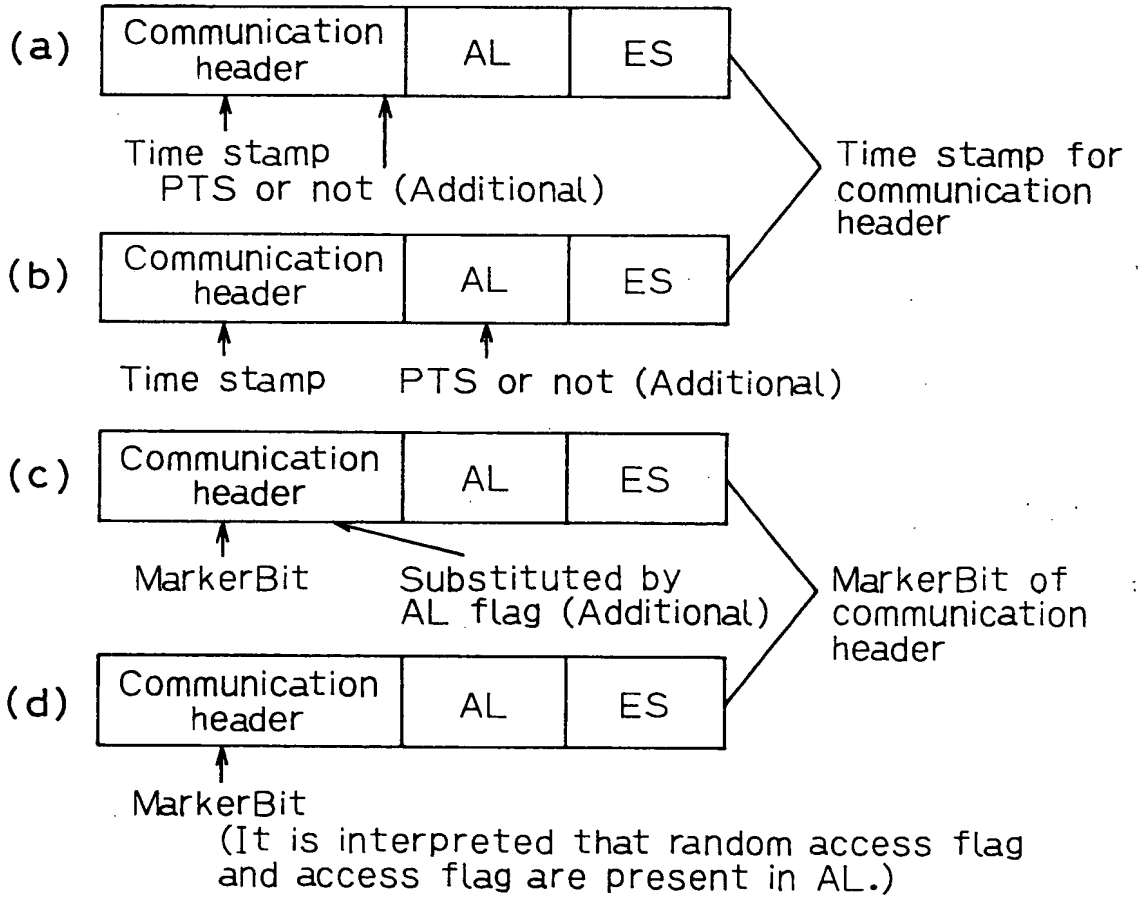


Fig. 5(a)

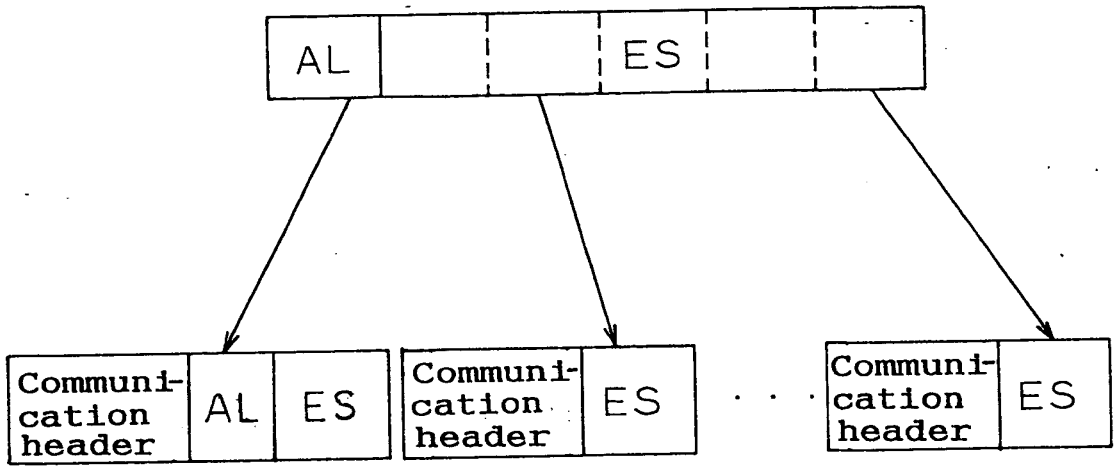

AL
ES
Communi-cation header
AL
ES
Communi-cation header
ES
Communi-cation header
ES

Fig. 5(b)

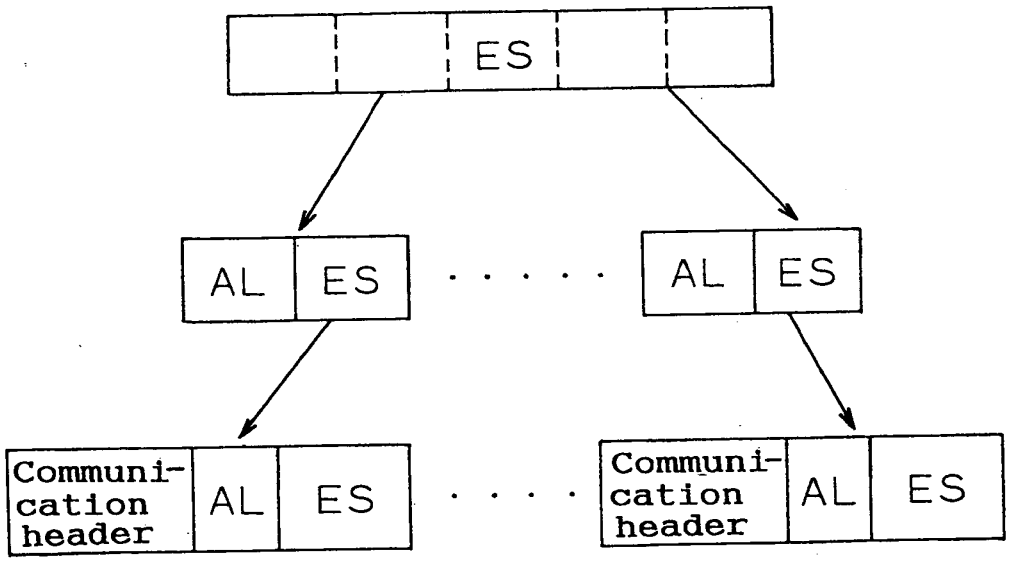
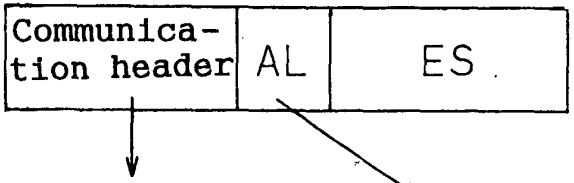

ES
AL
ES
AL
ES
Communi-cation header
AL
ES
Communi-cation header
AL
ES

Fig. 6(a)

Method for making the most use of RTP base



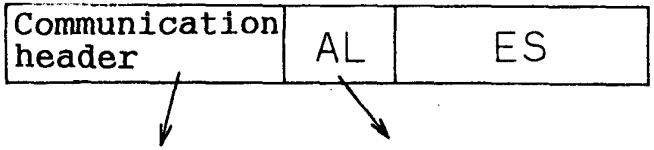
(RTP header: Communication header)

Sequence number
Time stamp
Payload type
MarkerBit
Object ID

Priority
Timer (Counter)
Time stamp is a flag showing PTS or not.

Fig. 6(b)

Method for simplifying the communication header



Sequence number

MarkerBit
Time stamp
Time stamp is a flag showing PTS or not.
Payload type
Object ID
Priority
Timer (Counter)

Fig. 6(c)

Method for changing every AL information to communication header

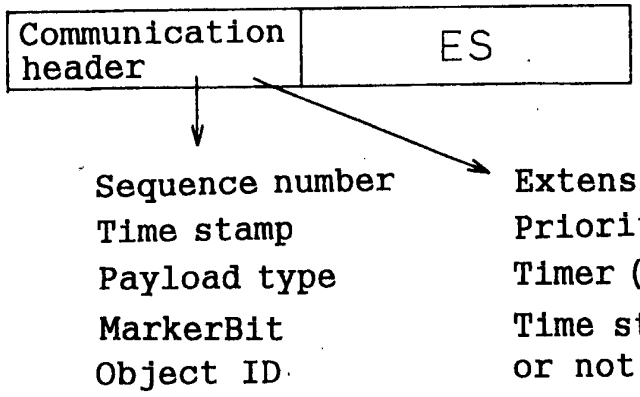


Fig. 6(d)

Method for transmitting as another packet

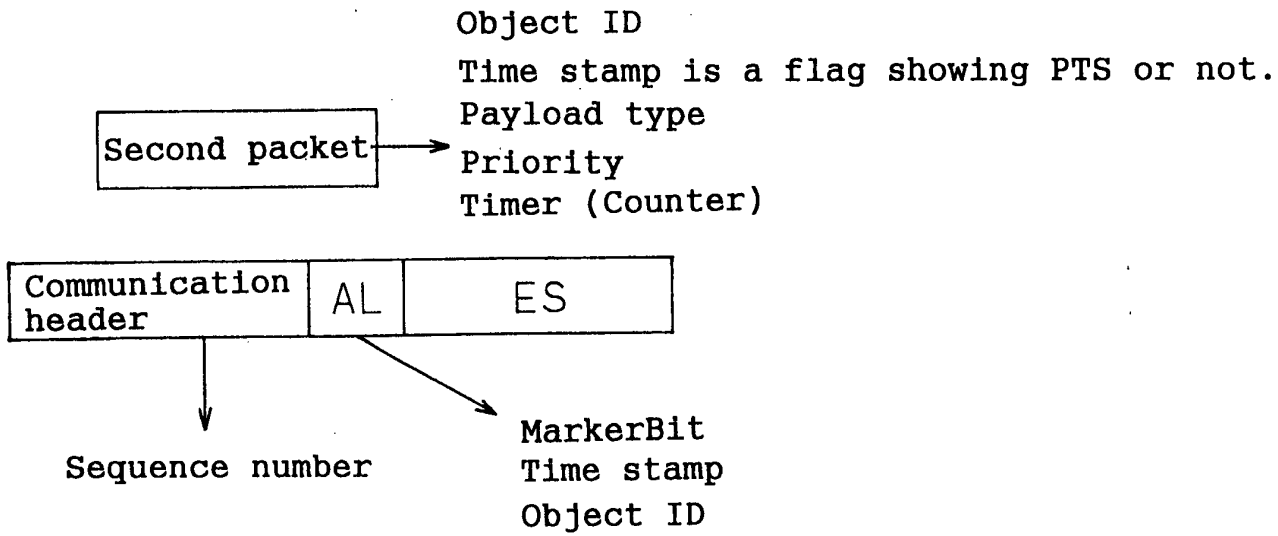
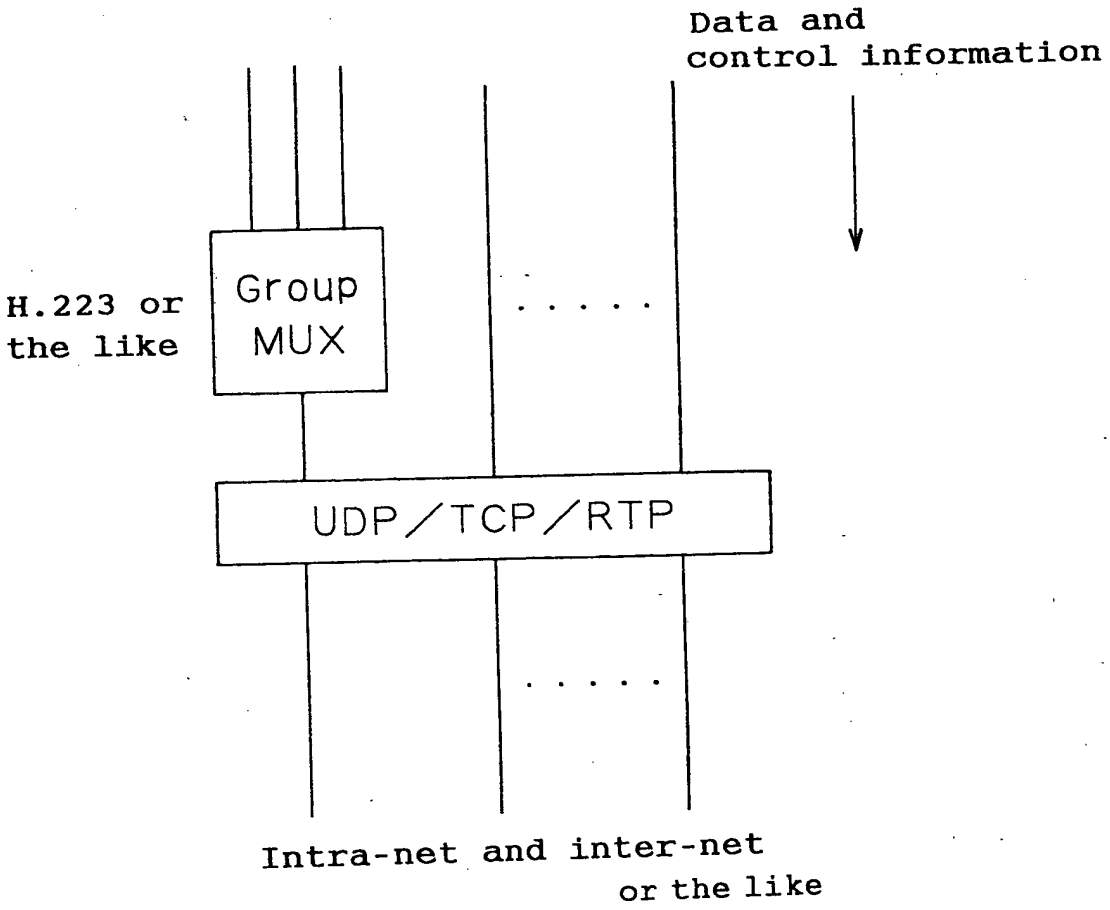
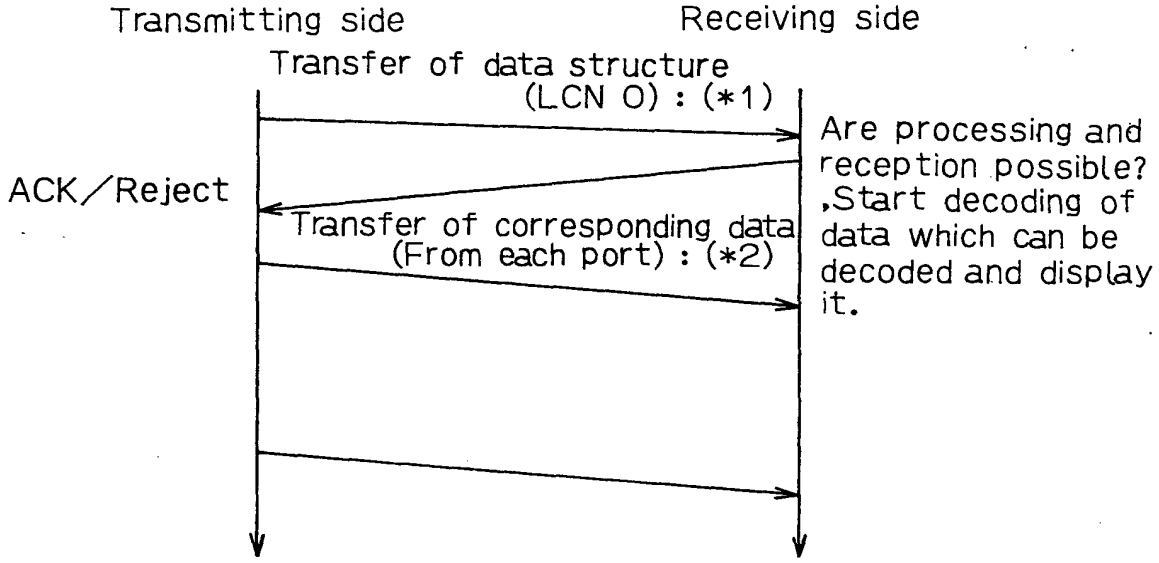


Fig. 7

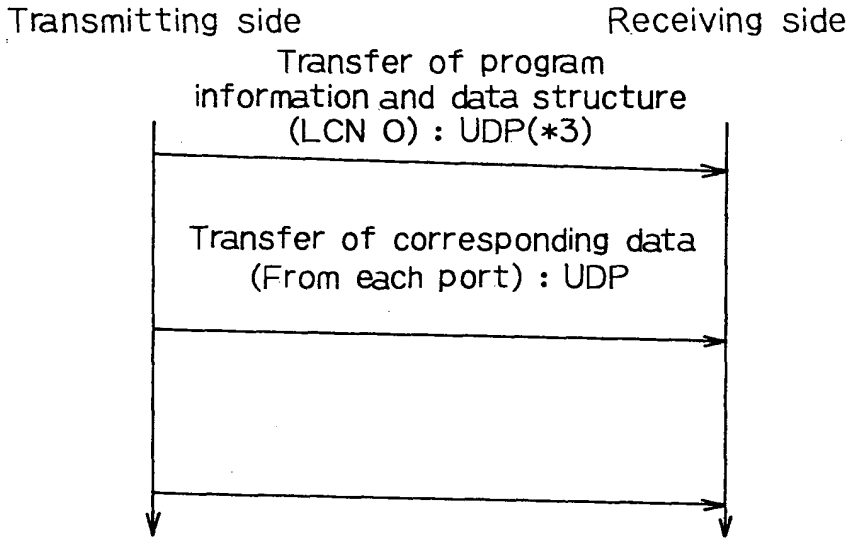

Data and
control information
H.223 or
the like
Group
MUX
UDP/TCP/RTP
Intra-net and inter-net
or the like

• Broadcast program transmitting procedure

〈Broadcast type and communication type including return channel〉



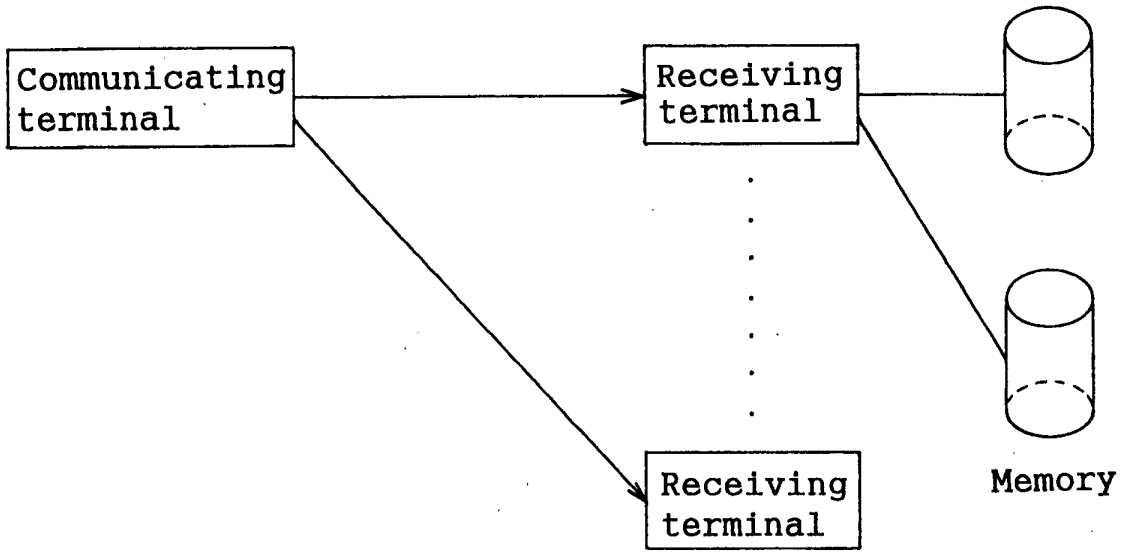
〈Broadcast type (with no return channel)〉



(*1) Must be a system for detecting and retransmitting a packet loss like TCP.
(*2) RTP/RTCP or TCP/IP
(*3) Same data (picture or sound) or control information (broadcast program or data structure) is continuously repeatedly transmitted. A packet is detected and sequence is kept at a receiving terminal in accordance with a sequence number. (To be used in a local closed region. Traffic becomes too large.)

Fig. 9(a)

When program or data is present at a receiving terminal



Program or data identifier to be required

Flag, counter, or timer for communicating a point of time to be required

Fig. 9(b)

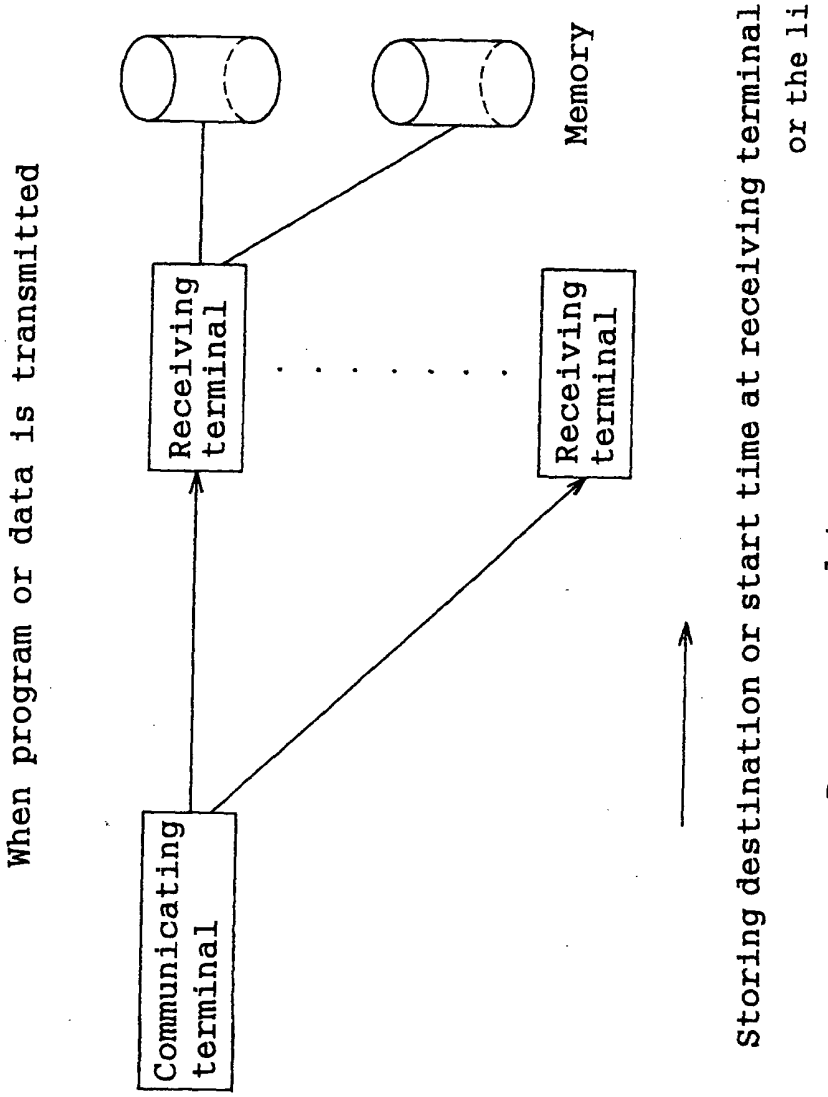

When program or data is transmitted
Communicating terminal
Receiving terminal
Receiving terminal
Memory
Storing destination or start time at receiving terminal or the like
Program or data

Fig. 10(a)

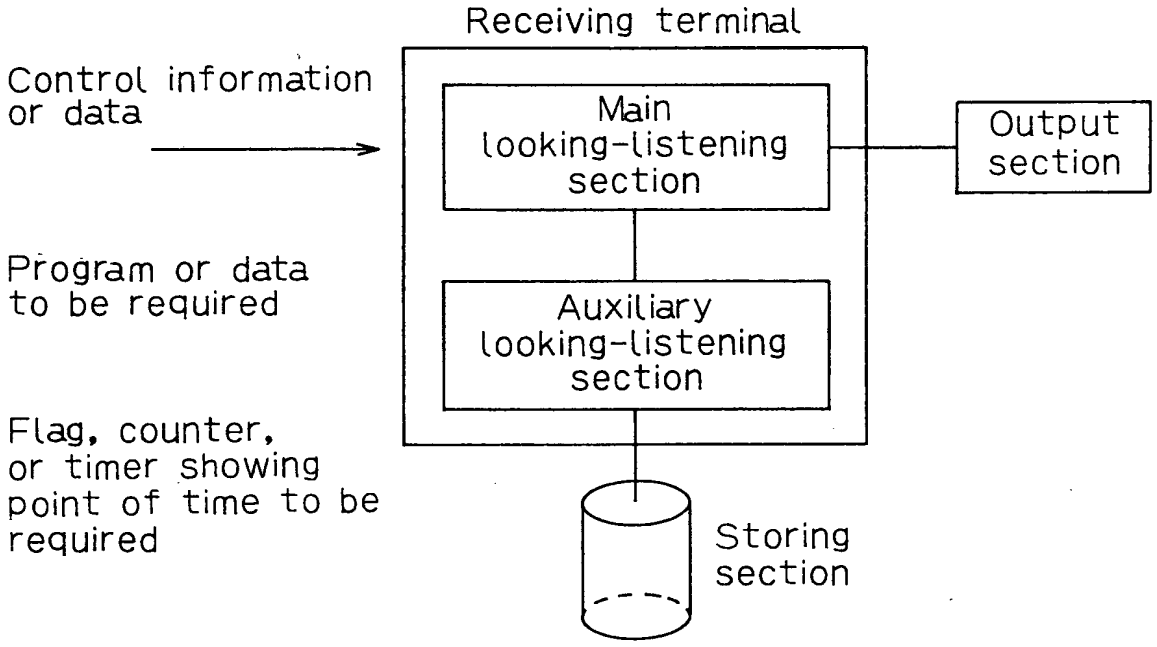


Fig. 10(b)

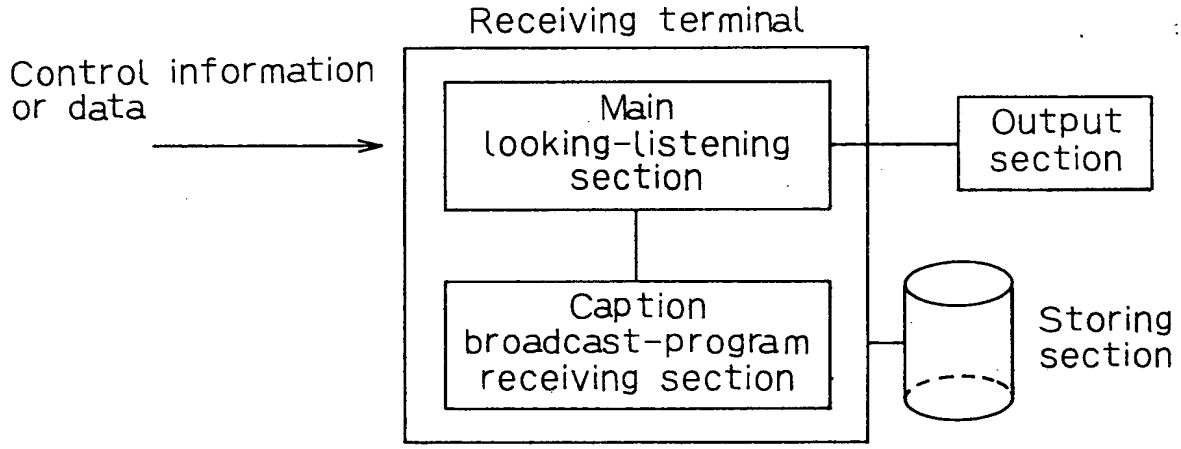
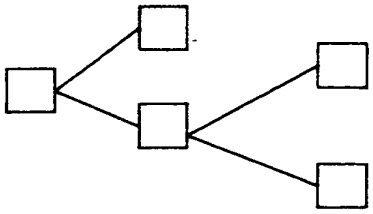


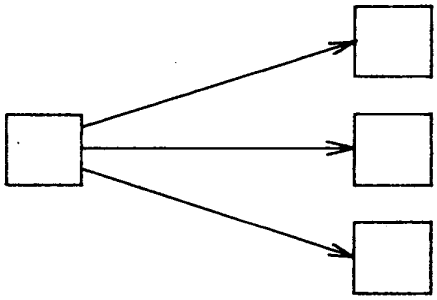
Fig. 11(a)

<Hierarchical image of object>



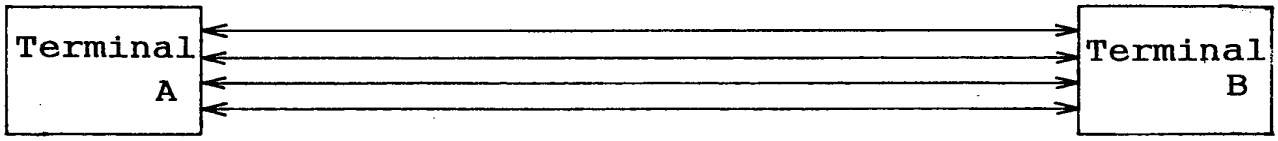
<Transmission image of object>

<1. Broadcast type>



<2. Communication type>

RTP/RTCP (Program ID of each logical channel is fixed.)



LCNO (control)

```
==========================================
=
--Capability exchange definitions(original from H.245)
==========================================
=
TerminalCapabilitySet                    ::=SEQUENCE
{
  sequenceNumber                         SequenceNumber,


  multiplexCapability                        MultiplexCapabilityOPTIONAL
  capabilityTable                        SET SIZE(1..256) OF CapabilityTableEntryOPTIONAL,
  capabilityDescriptors                  SET SIZE(1..256) OF CapabilityDescriptorOPTIONAL,
  mpeg4Capability                        MPEG4CapabilityOPTIONAL.
...
}
```

Fig. 12

```
-- MPEG4 Capability definitions

MPEG4Capability                    ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  NumberOfProcessObject            SEQUENCE
  {
      MaxNumberOfVideo             INTEGER(0..1023),
      ...
      MaxNumberOfSounds            INTEGER(0..1023),
      ...
      MaxNumberOfMux               INTEGER(0..1023),
  }
  reconfigurationALCapability      BOOLEAN,
  ...
}
MPEG4CapabilityAck                 ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  ...
}
MPEG4CapabilityReject              ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  NumberOfProcessObject            SEQUENCE
  {
      maxNumberOfVideo             MaxNumberOfVideo,
      ...
      maxNumberOfSounds            MaxNumberOfSounds
      ...
      MaxNumberOfMux               maxNumberOfMux,
  }
  reconfigurationALCapability      BOOLEAN,
  ...
}
```

Fig. 13(a)

```
-- ================================================
--
-- Group MUX definitions
-- ================================================
--
CreateGroupMux                  ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  GroupMuxID                    INTEGER(0..1023),
  lanportNumber                 LANPortNumber,
  ...
}
CreateGroupMuxAck               ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  ...
}
CreateGroupMuxReject            ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  cause                         CHOICE
  {
    ...
  }
  ...
}
```

Fig. 13(b)

```
DestoryGroupMux                 ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  GroupMuxID                    INTEGER(0..1023),
  ...
}
DestoryGroupMuxAck              ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  ...
}
DestoryGroupMuxReject           ::=SEQUENCE
{
  sequenceNumber                SequenceNumber,
  cause                         CHOICE
  {
     ...
  }
  ...
}
```

Fig. 13(c)

```
PortNumberStructure                ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  lanPortNumber                    LANPortNumber,
  numberOfLogicalNumber            INTEGER(1..15),
  SEQUENCE SIZE(1..15) OF PortStructureElement,
  ...
}
PortStructureElement               ::=SEQUENCE
{
  logicalPortNumber                LogicalPortNumber,
  ...
}
PortNumberStructureAck             ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  ...
}
PortNumberStructureReject          ::=SEQUENCE
{
  sequenceNumber                   SequenceNumber,
  cause                            CHOICE
  {
     ...
  }
  ...
}
```

Fig. 14

```
=============================================
=
   -Logical channel signalling definitions(original from H.245)
   -MPEG4 Object Create Operation(for LANPortNumber)
=============================================
=
OpenLogicalChannel                    ::=SEQUENCE
{
   fowardLogicalChannelNumber         LogicalChannelNumber,

   fowardLogicalChannelParameters     SEQUENCE
   {
     portNumber                       INTEGER(0..65535)OPTIONAL,
     dataType                         DataType,
     multiplexParameters              CHOICE
     {
      h222LogicalChannelParameters    H222LogicalChannelParameters,
      h223LogicalChannelParameters    H223LogicalChannelParameters,
      v76LogicalChannelParameters     v76LogicalChannelParameters,
      ...,
      h2250LogicalChannelParameters   H2250LogicalChannelParameters,
      h223AnnexALogicalChannelParameters
      H223AnnexALogicalChannelParameters
      MPEG4LogicalChannelParameters   MPEG4LogicalChanelParameters,
      ...
     },
     ...,
   },
   ...,
}
```

```
MPEG4LogicalChannelParameters    ::=SEQUENCE
{
    -H.225BASE                   INTEGER(0..65535),
    LANportNumber                INTEGER(0..255),
    ProgramID
    ProgramName                  OCTETSTRING(SIZE(128)),
    ...
}
BroadcastChannelProgram          ::=SEQUENCE
{
    sequenceNumber               SequenceNumber,
    numberOfChannelNumber        INTEGER(0..1023),
    SEQUENCE SIZE(1..1023) OF MPEG4LogicalChannelParameters
}
ChangeLogicalChannelAttribute    ::=SEQUENCE
{
    sequenceNumber               SequenceNumber
    lanportNumber                LANPortNumber,
    ProgramID                    INTEGER(0..255),
    ...
}
ChangeLogicalChannelAttributeAck ::=SEQUENCE
{
    sequenceNumber               SequenceNumber,
    ...
}
ChangeLogicalChannelAttributeReject ::=SEQUENCE
{
    sequenceNumber               SequenceNumber,
    cause                        CHOICE
    {
      ...
    }
    ...
}
```

Fig. 16(a)

```
==========================================
=
-MPEG4 Object Class definition
==========================================
MPEG4 Object Class definition        ::=SEQUENCE
{
    sequenceNumber                   SequenceNumber,
    ProgramID                        INTEGER(0..255),
    NumberOfObjectsList              INTEGER(0..1023),
    SEQUENCE SIZE(1..1023) OF ObjectStructureElement
}
ObjectStructureElement               ::=SEQUENCE
{
    SSRC                             INTEGER(0..16777215),
    LANPortNumber                    INTEGER(1024.5000),
O                                      --forRPT(Video&Sound)
    ScrambleFlag                     BOOLEAN,
CGDOffset                            INTEGER(0..255),
MediaType                            INTEGER(0..255),
...

}

MPEG4 Object Class definitionAck     ::=SEQUENCE
{
    sequenceNumber                   SequenceNumber,
    ...
}
MPEG4 Object Class definitionReject  ::=SEQUENCE
{
    sequenceNumber                   SequenceNumber,
    cause                            CHOICE
    {
      ...
    }
    ...
}
```

Fig. 16(b)

```
==========================================
=
-Adaptation Layer Reconfiguration Request definitions
==========================================
ALReconfiguration                    ::=CHOICE
{
  sequenceNumber                     SequenceNumber,
  RandomAccessFlagMaxBit             INTEGER(0...2),
  PresentationTimeStampsMaxBit       INTEGER(0...32),
  CGDPriorityMaxBit                  INTEGER(0...8),
                                        --forVideo and Sound
  ...
}
==========================================
=
-Adaptation Layer Reconfiguration Response definitions
==========================================
ALReconfigurationAck                 ::=SEQUENCE
{
  sequenceNumber                     SequenceNumber,
  ...
}
ALReconfigurationReject              ::=SEQUENCE
{
  sequenceNumber                     SequenceNumber,
  cause                              CHOICE
  {
     ...
  }
  ...
}
```

<Relation between AL, ES, and RTP>

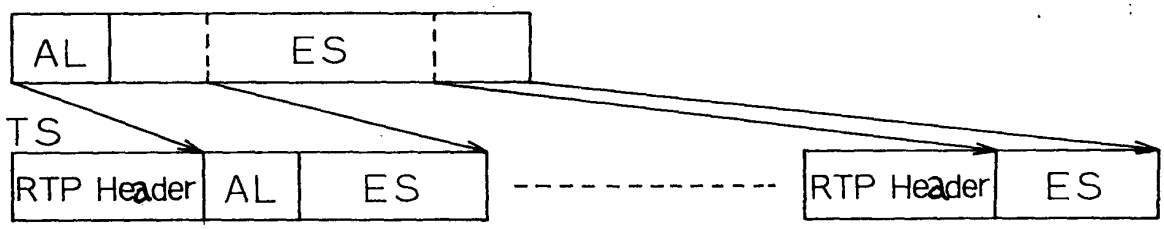


Fig. 17

```
==========
=
-Setup Program and Data Request definitions
==========
Setup Request                          ::=CHOICE
{
  sequenceNumber                       SequenceNumber,
  SSRC IMEGER(0..16777215)2^32,
  Logical Channel Number,              INTEGER(1024...5000),
  setupitem                            CHOICE
  {
     executeProgramNumber              INTEGER(0...255),
     dataNumber                        INTEGER(0...255),
     executeCommandNumber              INTEGER(0...255),
  },
  nofitycounter                        CHOICE
  {
     flag                              BOOLEAN
     counter                           INTEGER(0...255),
     timer                             INTEGER(0...255),
  },
  ...,
}
```

Fig. 18

```
================================================
=
-control and AL attribute definitions
================================================

ControlALdefinition                    ::=CHOICE
{

  sequenceNumber                       SequenceNumber,
  AL                                   CHOICE
  {
     RandomAccessFlagUse               BOOLEAN,
     PresentationTimeStampUse          BOOLEAN,
     CGDPriorityUse                    BOOLEAN,
     ...
  },
```

Fig. 19(a)

```
classES_header{
    uint(4)    headerID;
    uint(24)   bufferSizeES;
    uint(1)    useTimeStamps;
    .......
    .......
    uint(16)   sequenceNumberMaxBit;
    uint(1)    useHeaderExtension;
    if (useHeaderExtension){
                uint(1)   accessUintStartFlag;
                uint(1)   randomAccessPointFlag;
                uint(1)   OCRsetFlag;
                uint(4)   degradationPriorityMaxBit;
    }
    uint(3)    reserved:
}
```

Fig. 19(b)

```
--Adaptation Layer PDU header configuration Request and Command definition
AL configuration                        ::=SEQUENCE
{
    sequenceNumber                      SequenceNumber,
    defaultHeaderConfiguration          BOOLEAN,
    headerID                            INTEGER(0..4),
    MPEG4ALPDUHeaderConfig              SEQUENCE
    {
      accessUintStartFlag               BOOLEAN,
      randomAccessPointFlag             BOOLEAN,
      OCRsetFlag                        BOOLEAN,
      degradationPriorityMaxBit         INTEGER(0..4),
      ...
    }
}
```

Fig. 20(a)

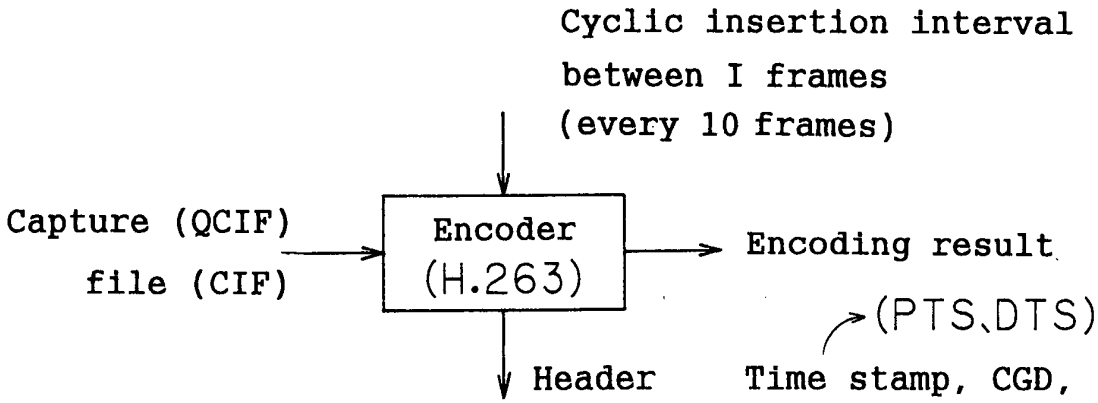


Fig. 20(b)

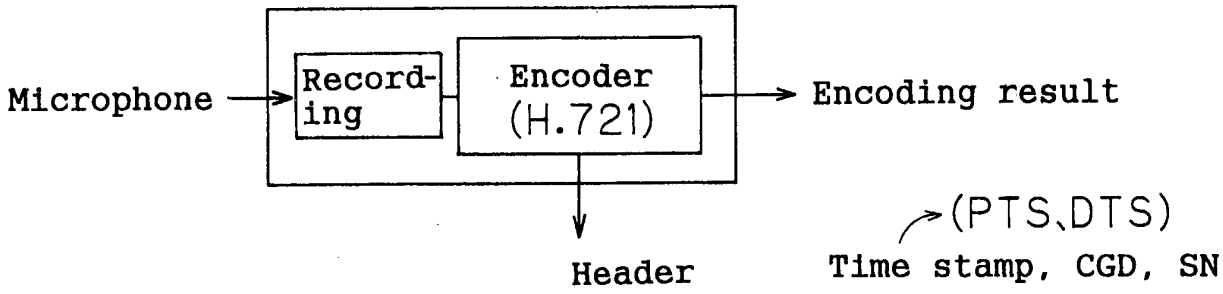


Fig. 20(c)

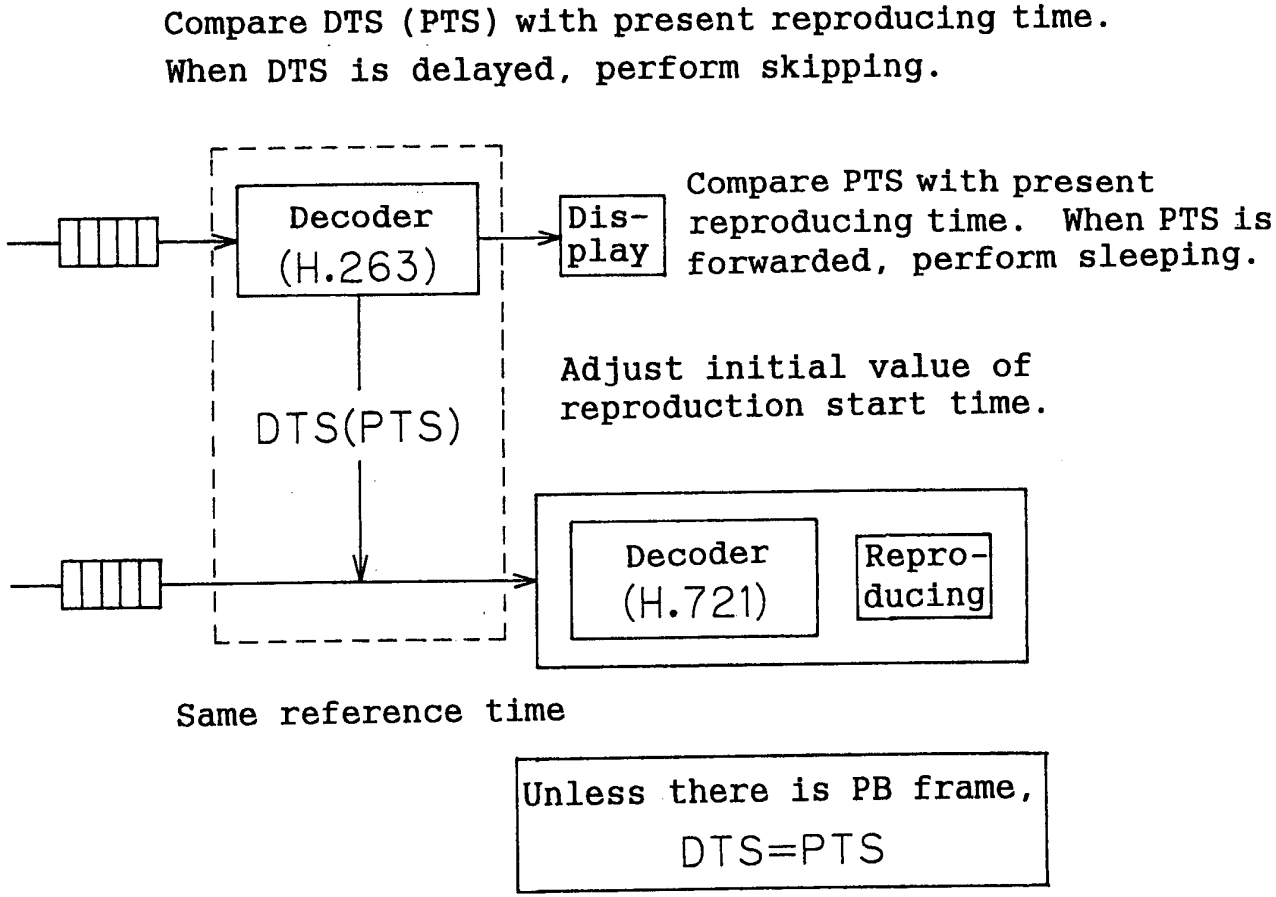

Compare DTS (PTS) with present reproducing time.
When DTS is delayed, perform skipping.
Decoder
(H.263)
Dis-
play
Compare PTS with present reproducing time. When PTS is forwarded, perform sleeping.
DTS(PTS)
Adjust initial value of reproduction start time.
Decoder
(H.721)
Repro-
ducing
Same reference time
Unless there is PB frame,
DTS=PTS

Fig. 21

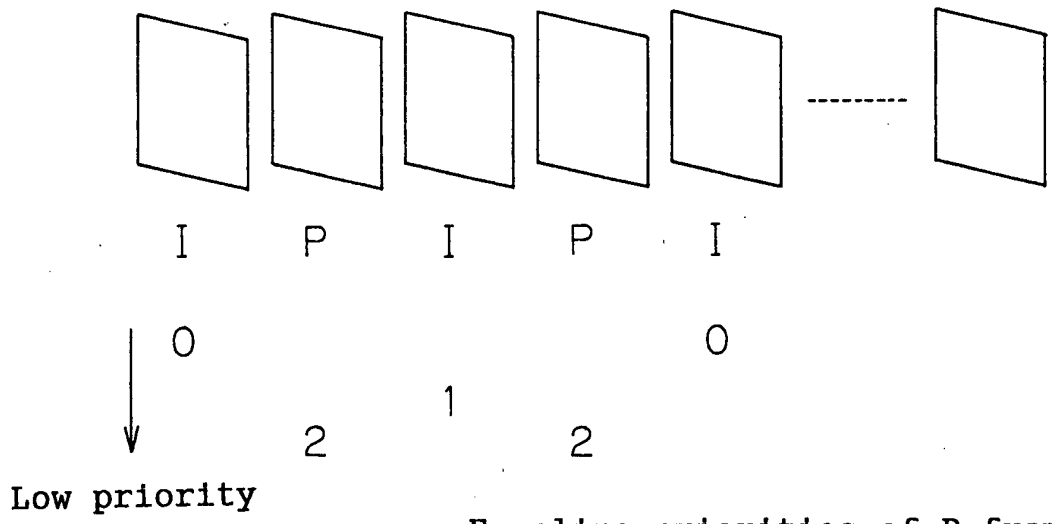

I
P
I
P
I
0
0
1
2
2
Low priority
Equalize priorities of P frames each other and those of PB frames each other.

Fig. 22

Processing at receiving terminal under overload(Common to dynamic picture and sound)
Thread for processing sound at system level is previously set it's processing priority to a value higher than that of thread for processing picture.

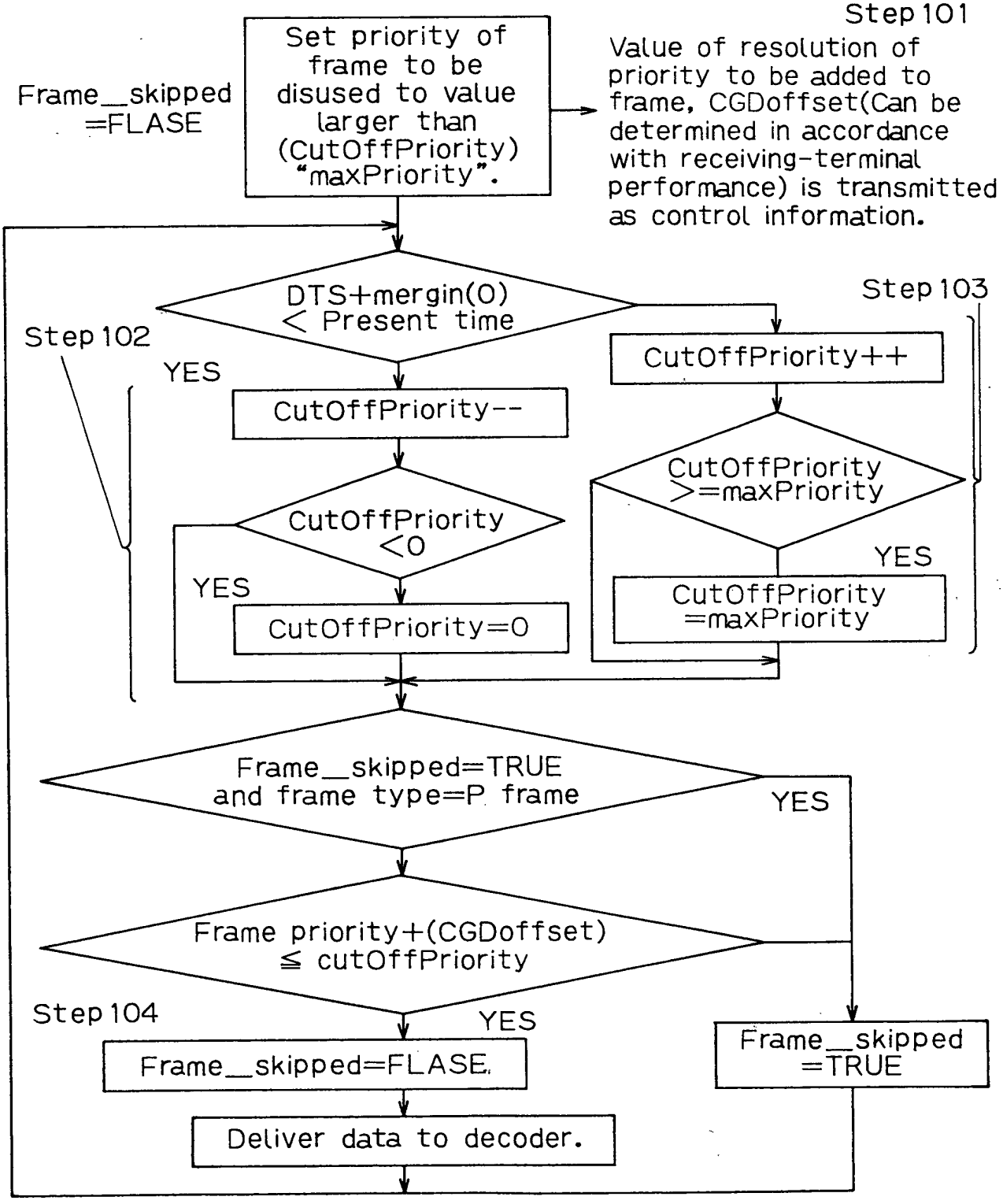
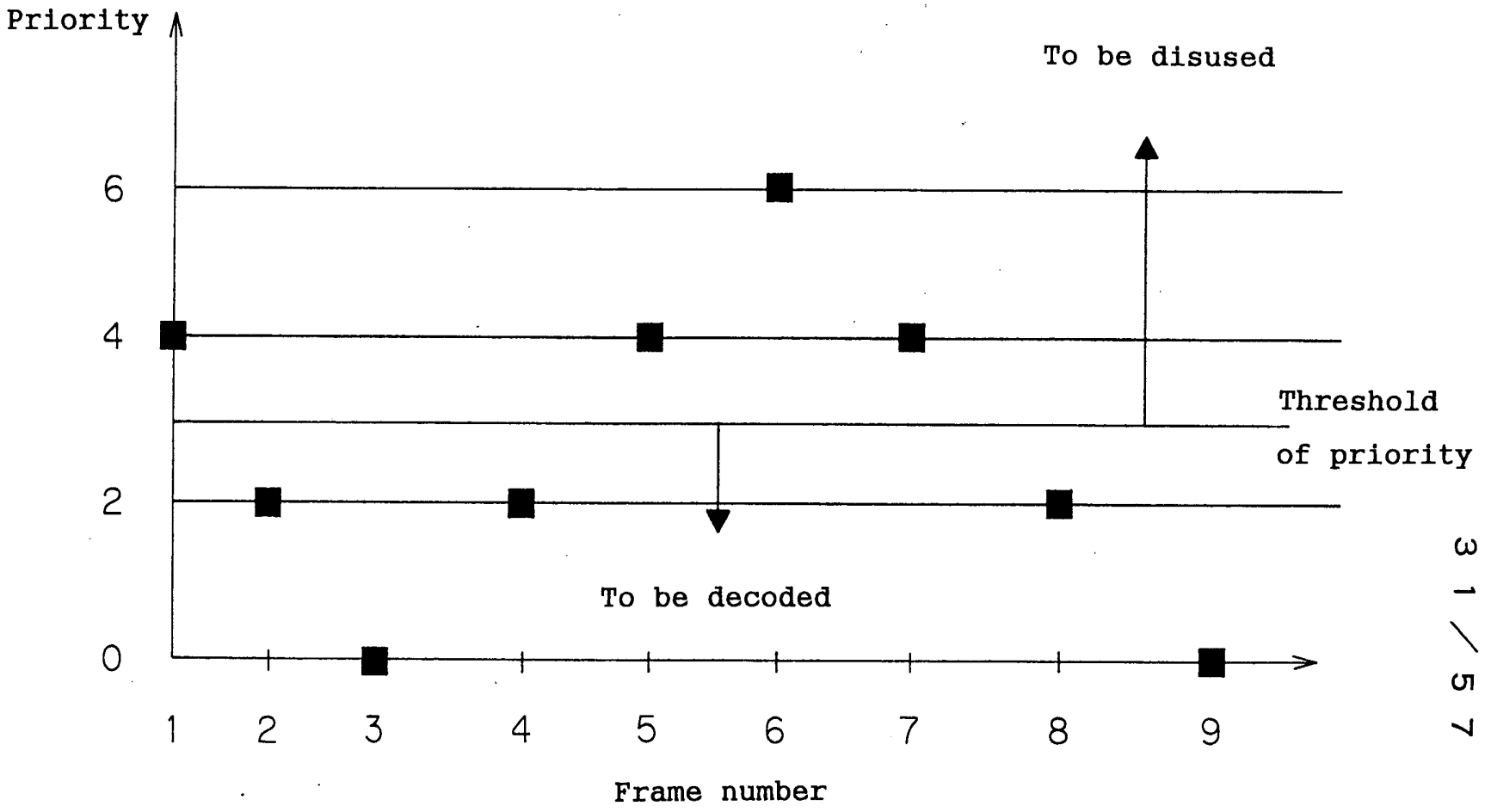


Fig. 23


Priority
To be disused
6
4
Threshold
of priority
2
To be decoded
0
1
2
3
4
5
6
7
8
9
Frame number

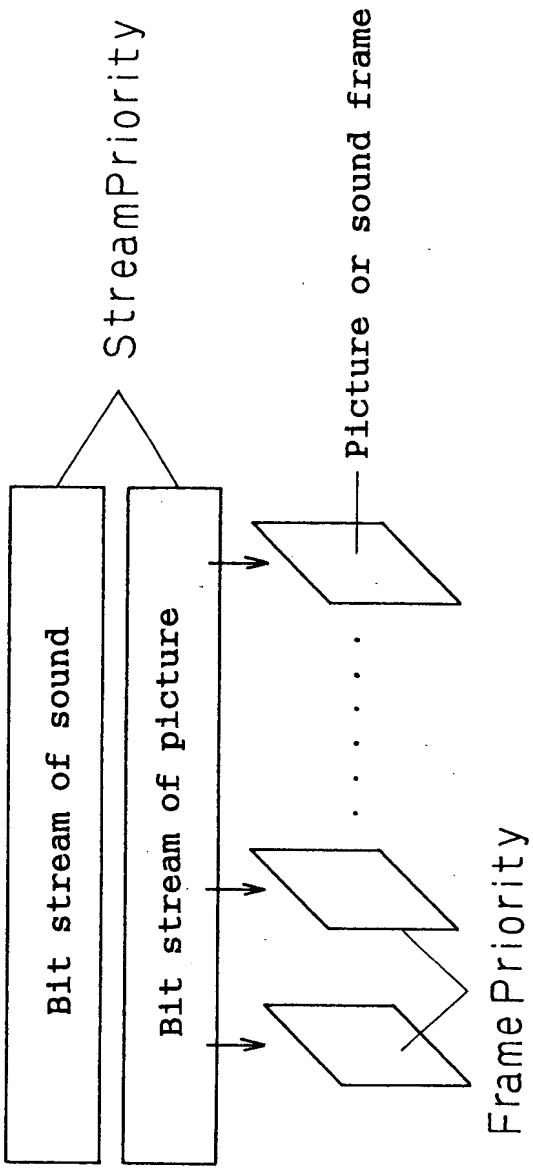

StreamPriority
Bit stream of sound
Bit stream of picture
Picture or sound frame
FramePriority

Fig. 24

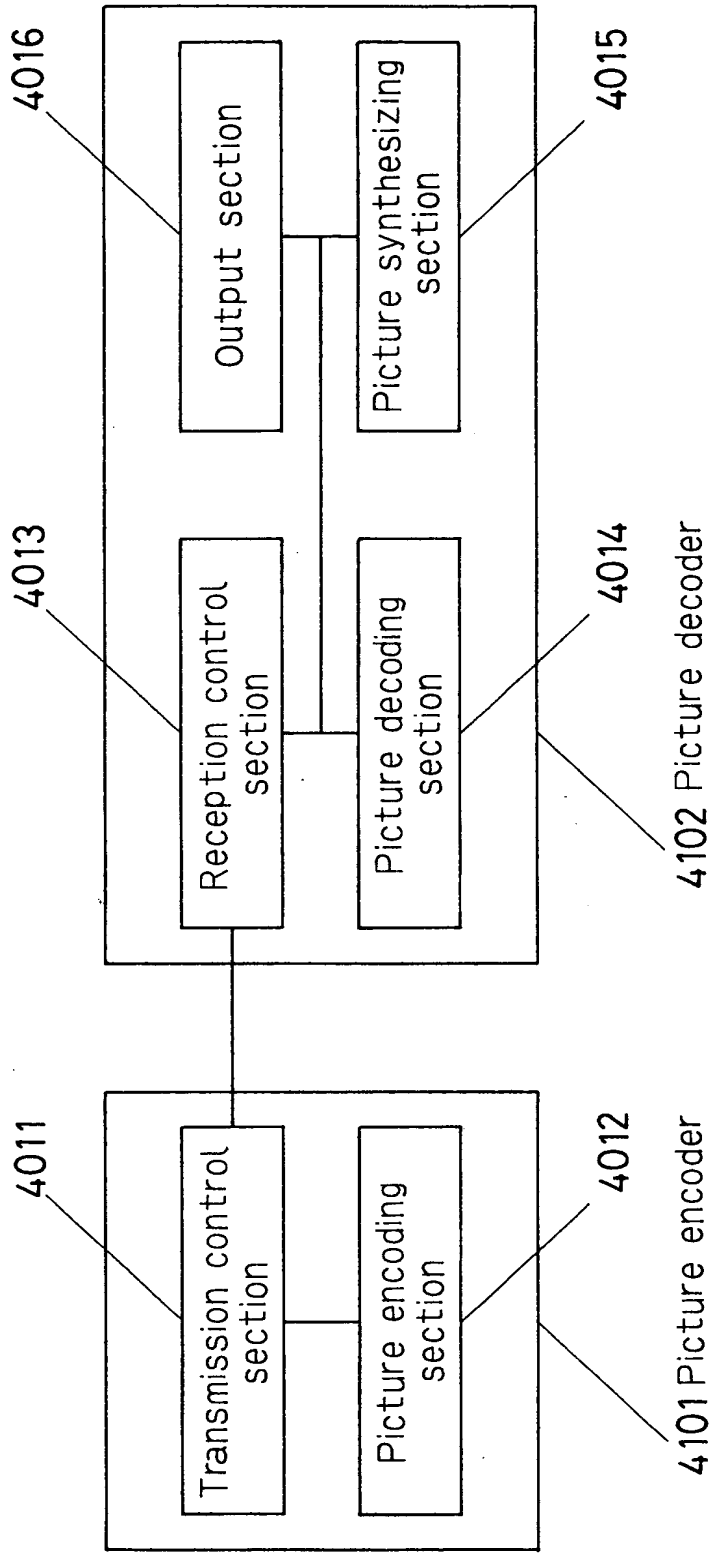

4016
Output section
4015
Picture synthesizing section
4013
Reception control section
4014
Picture decoding section
4102 Picture decoder
4011
Transmission control section
4012
Picture encoding section
4101 Picture encoder

Fig. 25

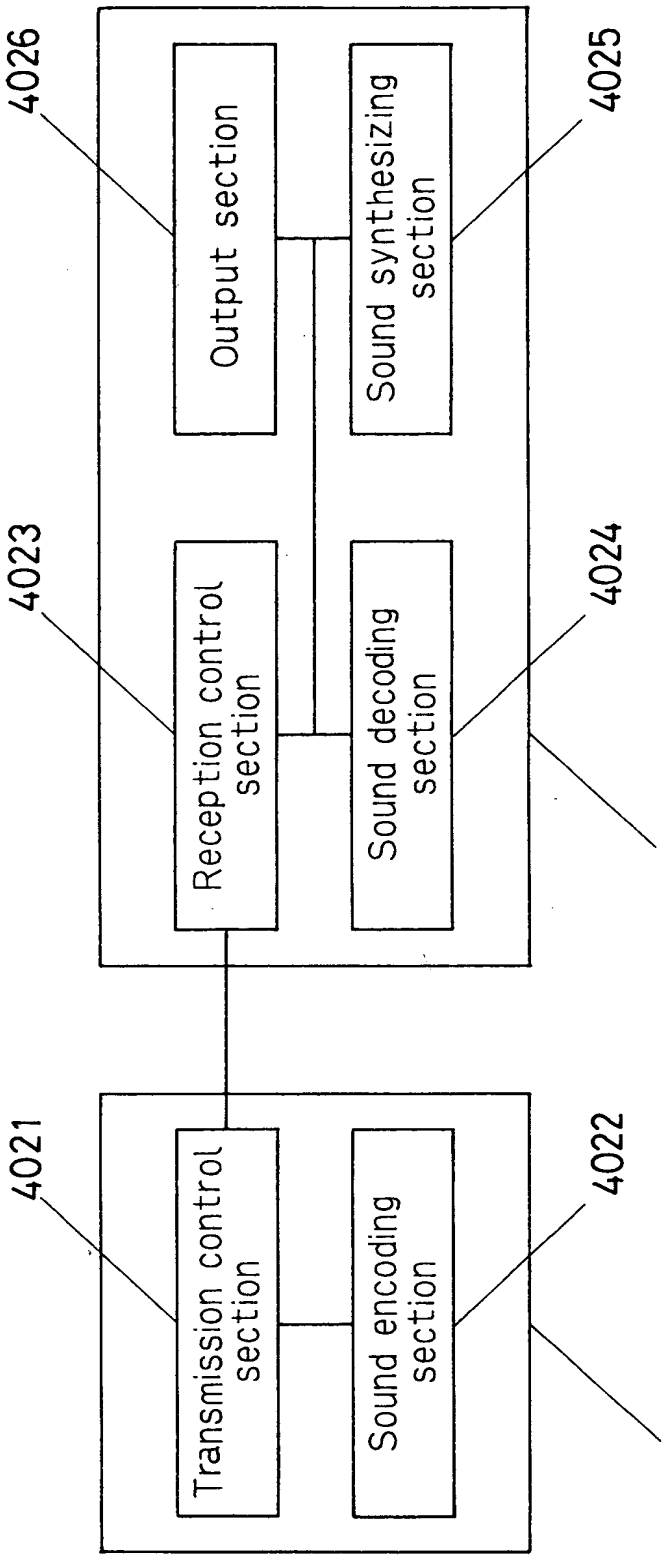

4026
Output section
4025
Sound synthesizing section
4023
Reception control section
4024
Sound decoding section
4202 Sound decoder
4021
Transmission control section
4022
Sound encoding section
4201 Sound encoder

Fig. 26

Fig. 27(a)

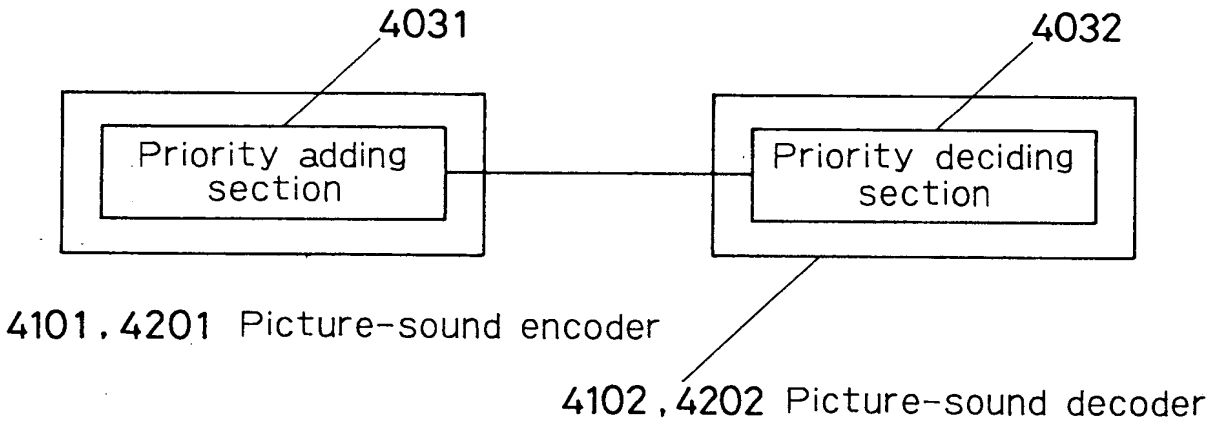


Fig. 27(b)

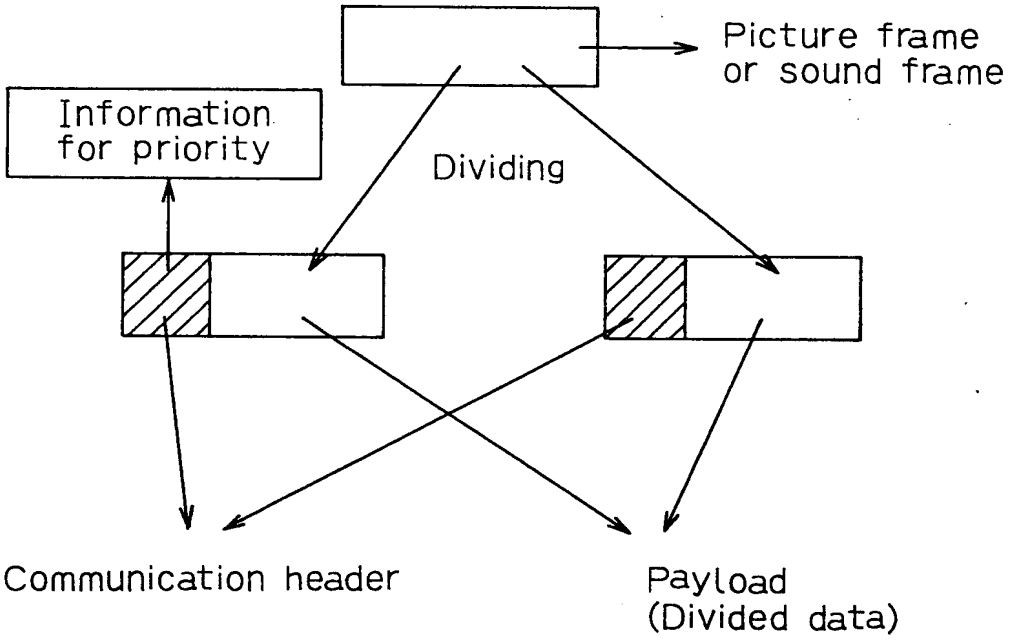


Fig. 28(a)

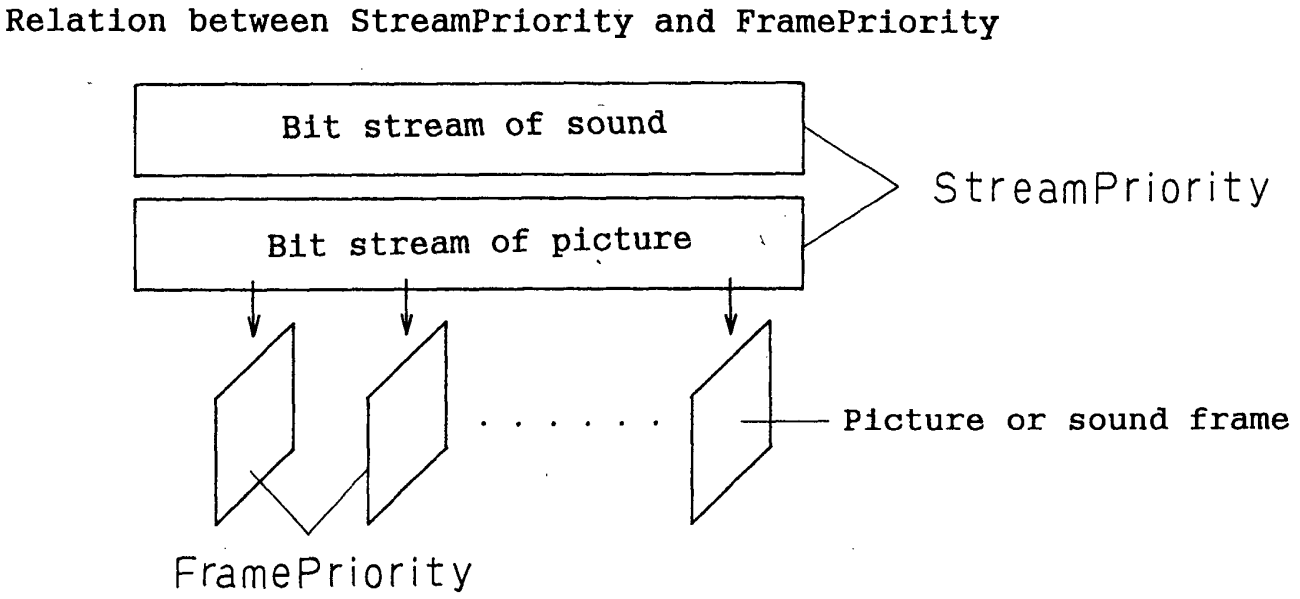

Relation between StreamPriority and FramePriority
Bit stream of sound
Bit stream of picture
StreamPriority
Picture or sound frame
FramePriority

Fig. 28(b)

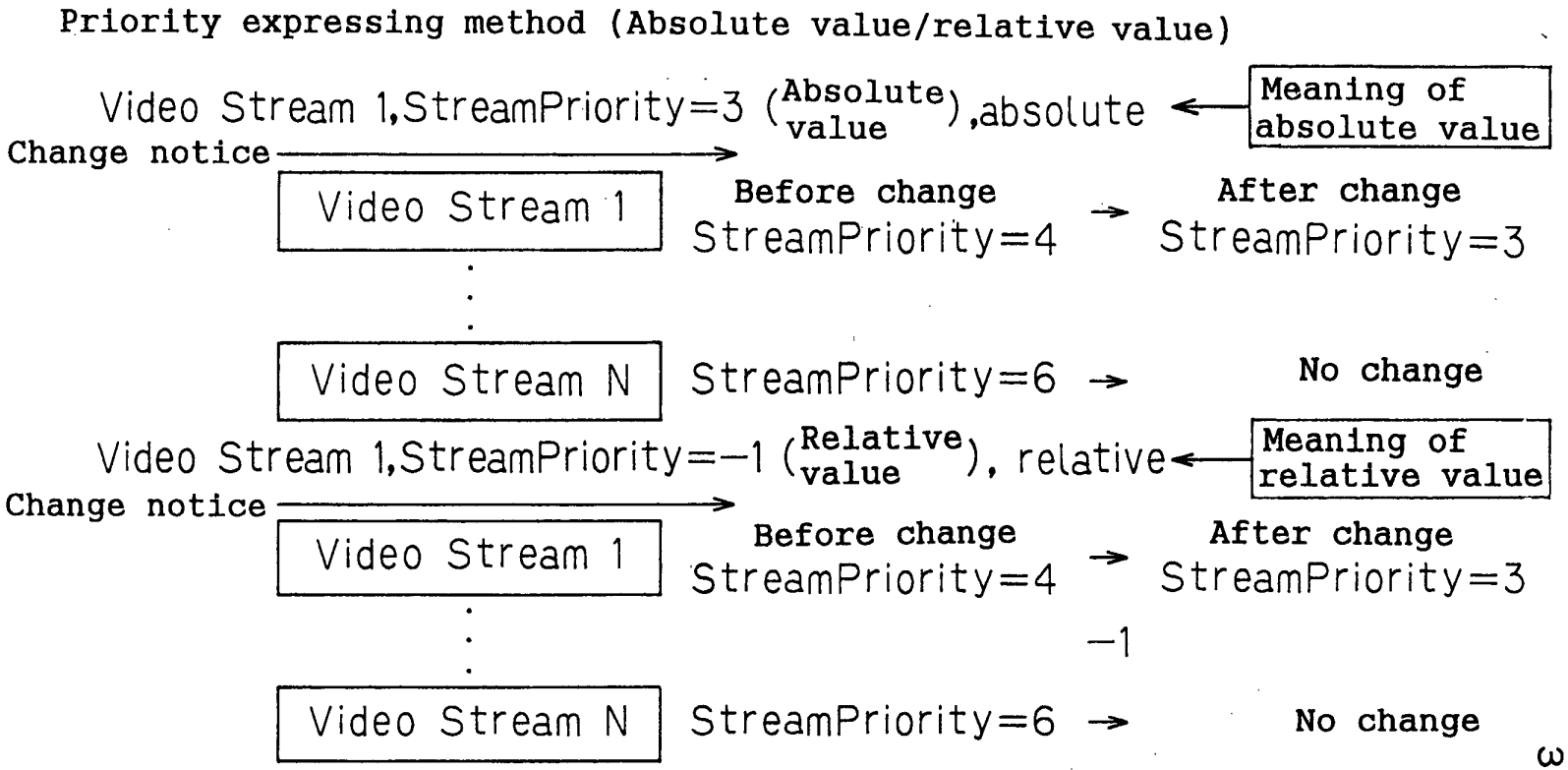

Priority expressing method (Absolute value/relative value)
Video Stream 1,StreamPriority=3 (Absolute value),absolute
Meaning of absolute value
Change notice
Video Stream 1
Before change
StreamPriority=4
After change
StreamPriority=3
Video Stream N
StreamPriority=6
No change
Video Stream 1,StreamPriority=−1 (Relative value), relative
Meaning of relative value
Change notice
Video Stream 1
Before change
StreamPriority=4
After change
StreamPriority=3
−1
Video Stream N
StreamPriority=6
No change

Fig. 28(c)

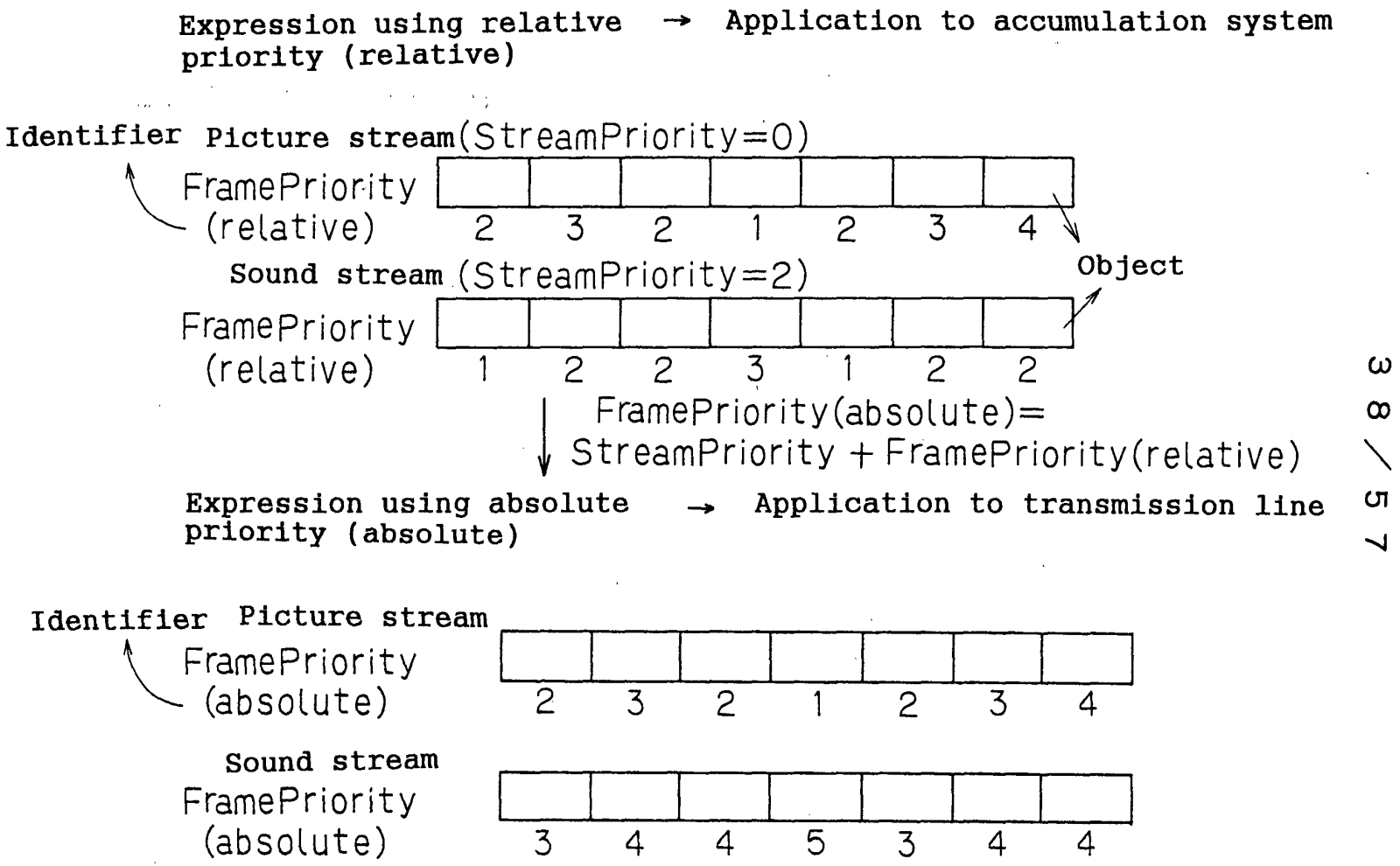

Expression using relative priority (relative)
→ Application to accumulation system
Identifier
Picture stream (StreamPriority=0)
FramePriority (relative)
2 3 2 1 2 3 4
Object
Sound stream (StreamPriority=2)
FramePriority (relative)
1 2 2 3 1 2 2
FramePriority(absolute)=
StreamPriority + FramePriority(relative)
Expression using absolute priority (absolute)
→ Application to transmission line
Identifier
Picture stream
FramePriority (absolute)
2 3 2 1 2 3 4
Sound stream
FramePriority (absolute)
3 4 4 5 3 4 4

Fig. 29

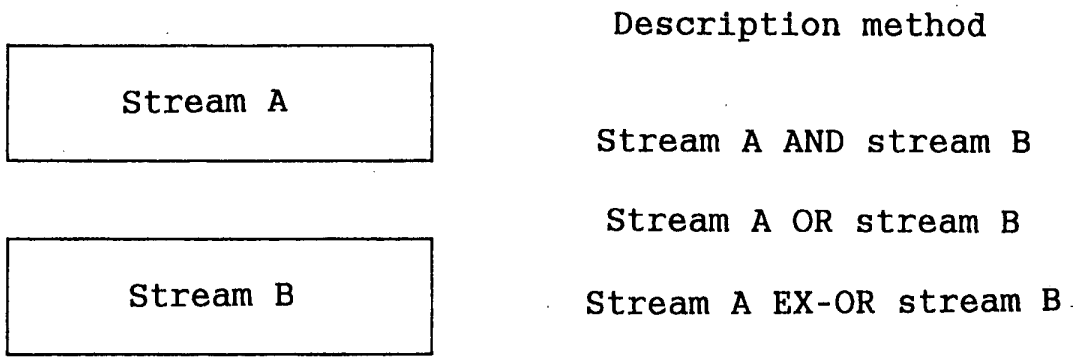

Description method
Stream A
Stream A AND stream B
Stream A OR stream B
Stream B
Stream A EX-OR stream B

Fig. 30

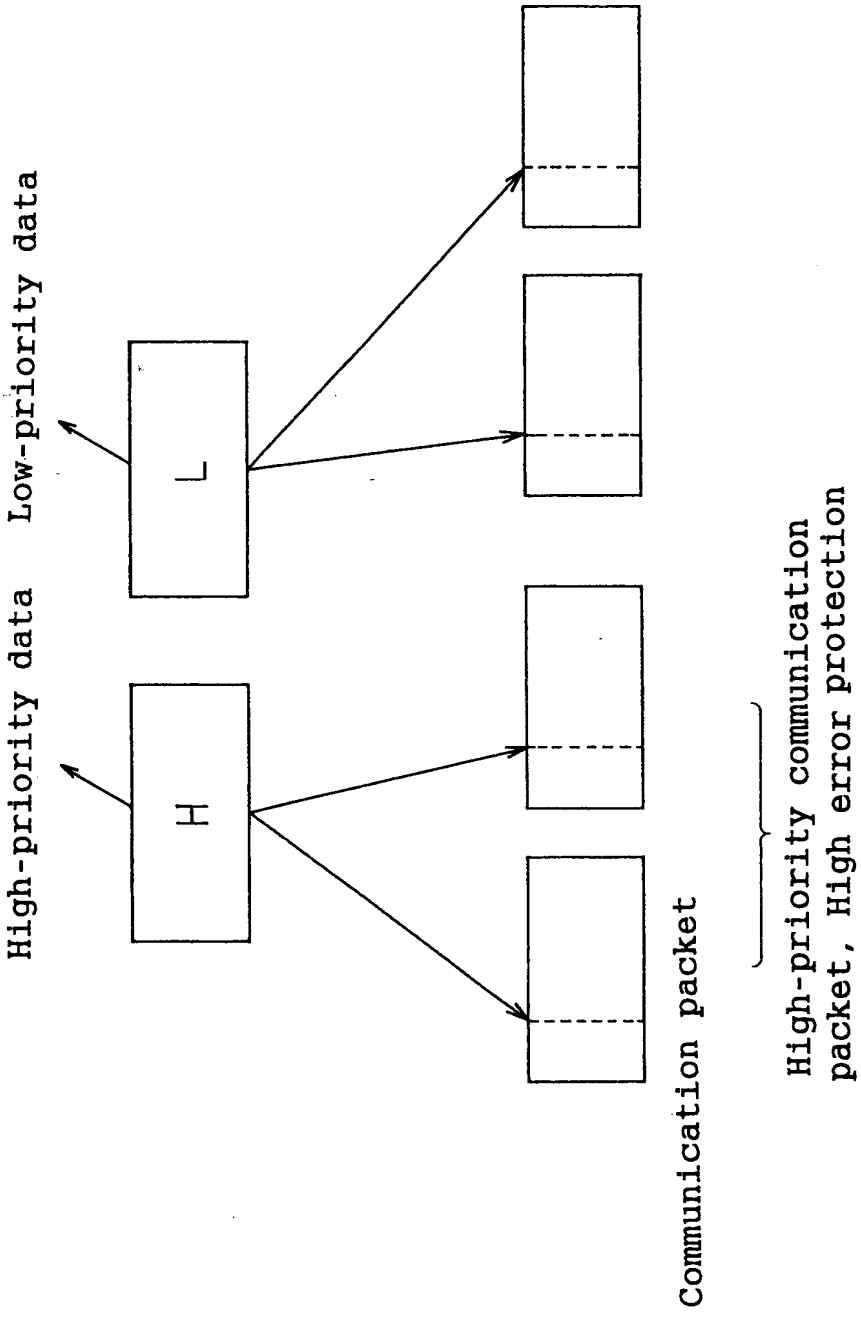
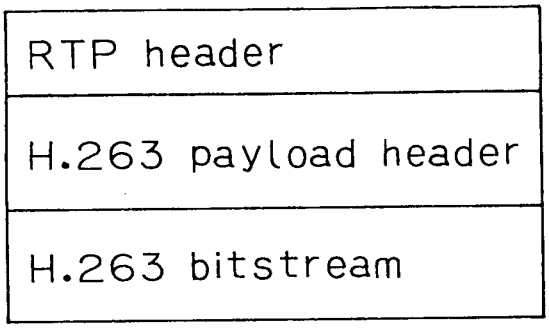

High-priority data
Low-priority data
H
L
Communication packet
High-priority communication packet, High error protection

Fig. 31



- Mode A: GOB, picture boundary

  Presence or absence of mode or PB, start and end positions of bit stream, and execution timing states of options of resolution, frame type, and H.263 → Core information

  DBQUANT, TR(for B frame), TR(for P frame) → To be set when PB frame is present

- Mode B: MB boundary without PB

  Core information for Mode A

  Information for quantization value (GQUANT), GOB number, absolute address of first MB in GOB, and movement vector (Horizontal and vertical directions)

- Mode C: MB boundary with PB

  Information for Mode B

  DBQUANT, TR(for B frame), TR(for P frame)

Relating of communication payload

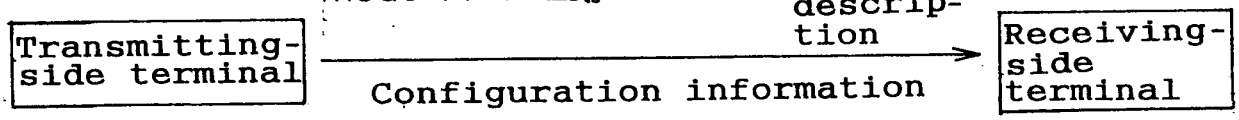
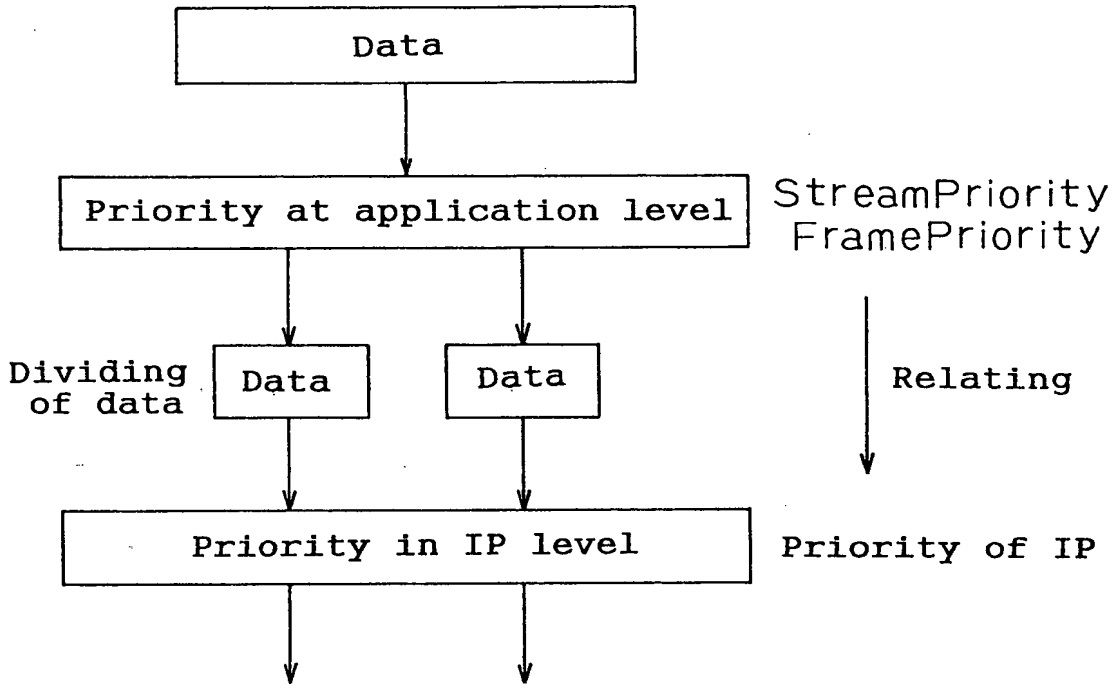


Fig. 32



Priority in data				Available range
StreamPriority	0	~	3	[0···15]
FramePriority	0	~	5	[0···15]
IPV6	8	~	15	[8···15]

Mapping of part

(Lowest) (Maximum priority)

Fig. 33

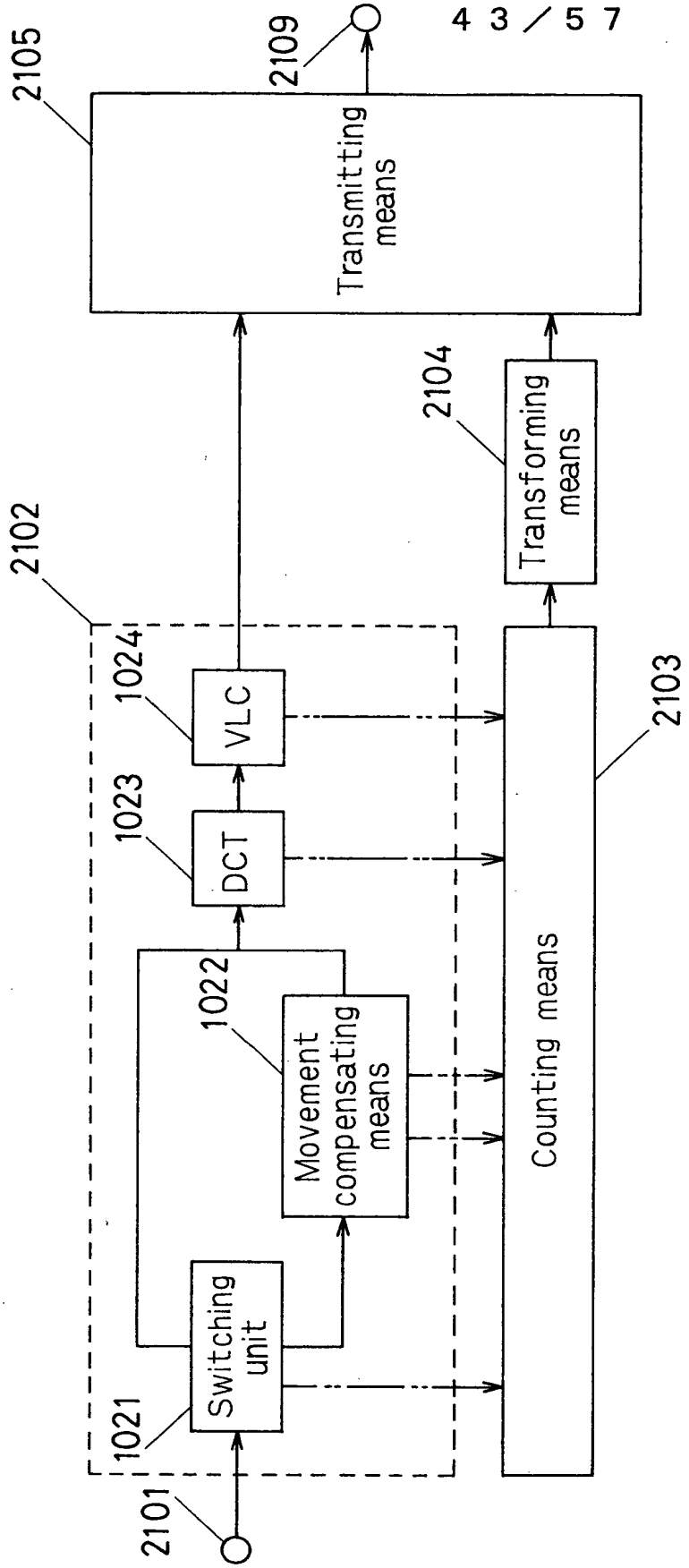

2105
2109
Transmitting means
2104
Transforming means
2102
1024
VLC
1023
DCT
1022
Movement compensating means
2103
Counting means
1021
Switching unit
2101

Fig. 34

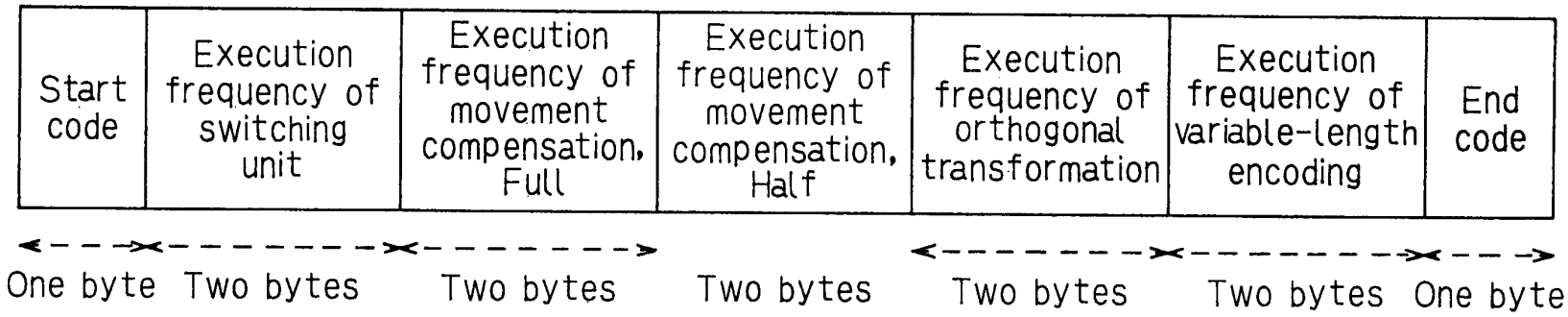

Start code
Execution frequency of switching unit
Execution frequency of movement compensation, Full
Execution frequency of movement compensation, Half
Execution frequency of orthogonal transformation
Execution frequency of variable-length encoding
End code
One byte
Two bytes
Two bytes
Two bytes
Two bytes
Two bytes
One byte

Fig. 35

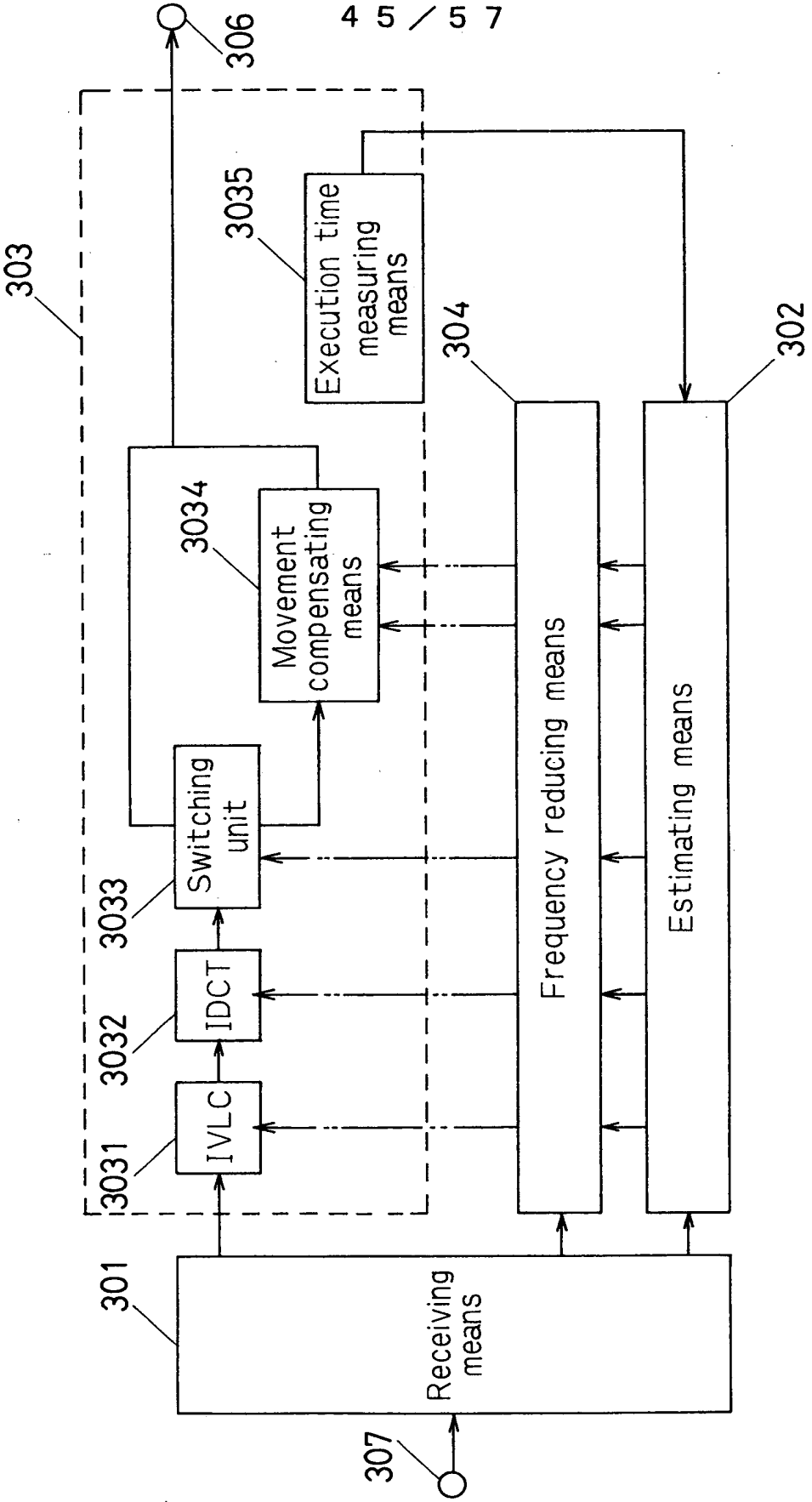

306
303
3035
Execution time measuring means
304
302
3034
Movement compensating means
Switching unit
3033
IDCT
3032
IVLC
3031
Frequency reducing means
Estimating means
301
Receiving means
307

Fig. 36

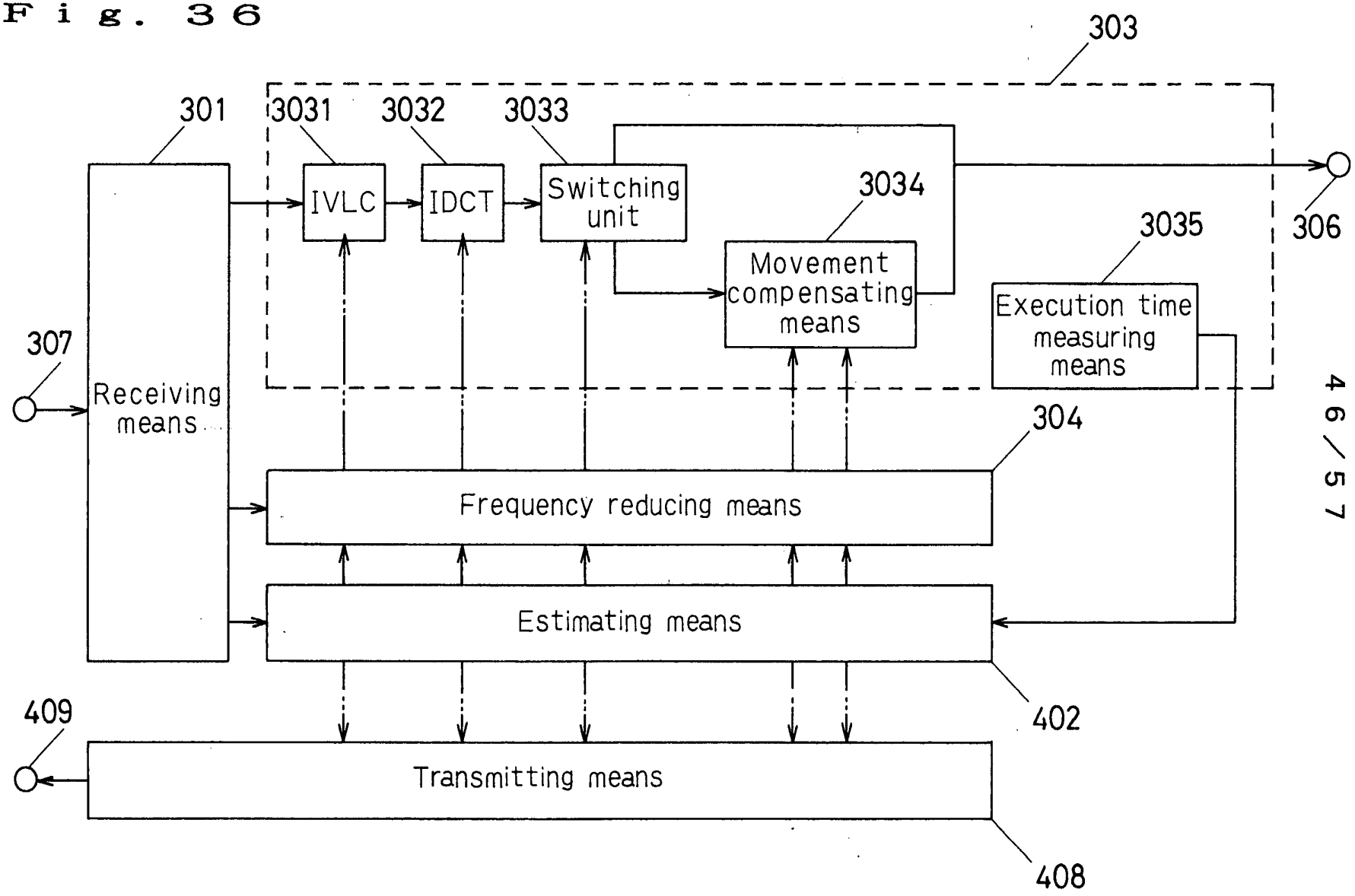
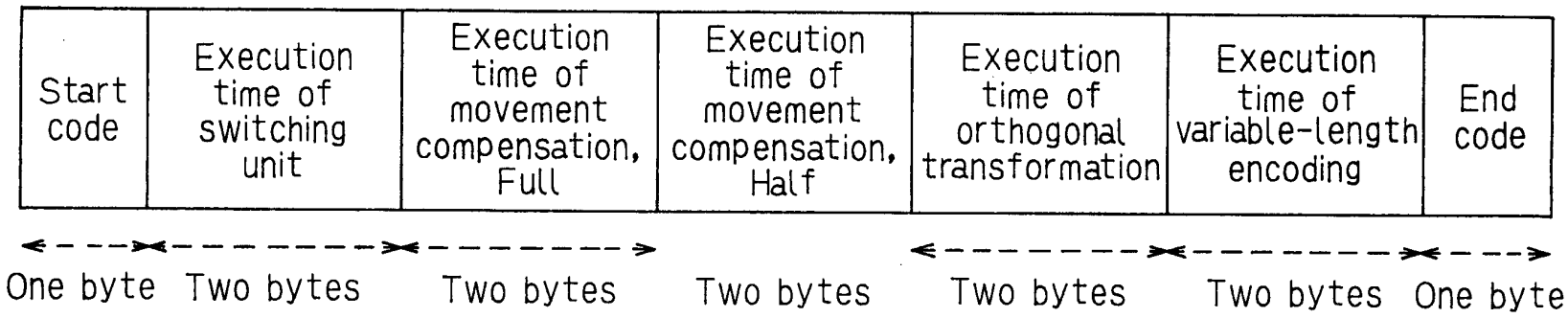

303
301
3031
3032
3033
IVLC
IDCT
Switching unit
3034
Movement compensating means
3035
Execution time measuring means
306
307
Receiving means
304
Frequency reducing means
Estimating means
402
409
Transmitting means
408

Fig. 37


Start code
Execution time of switching unit
Execution time of movement compensation, Full
Execution time of movement compensation, Half
Execution time of orthogonal transformation
Execution time of variable-length encoding
End code
One byte
Two bytes
Two bytes
Two bytes
Two bytes
Two bytes
One byte

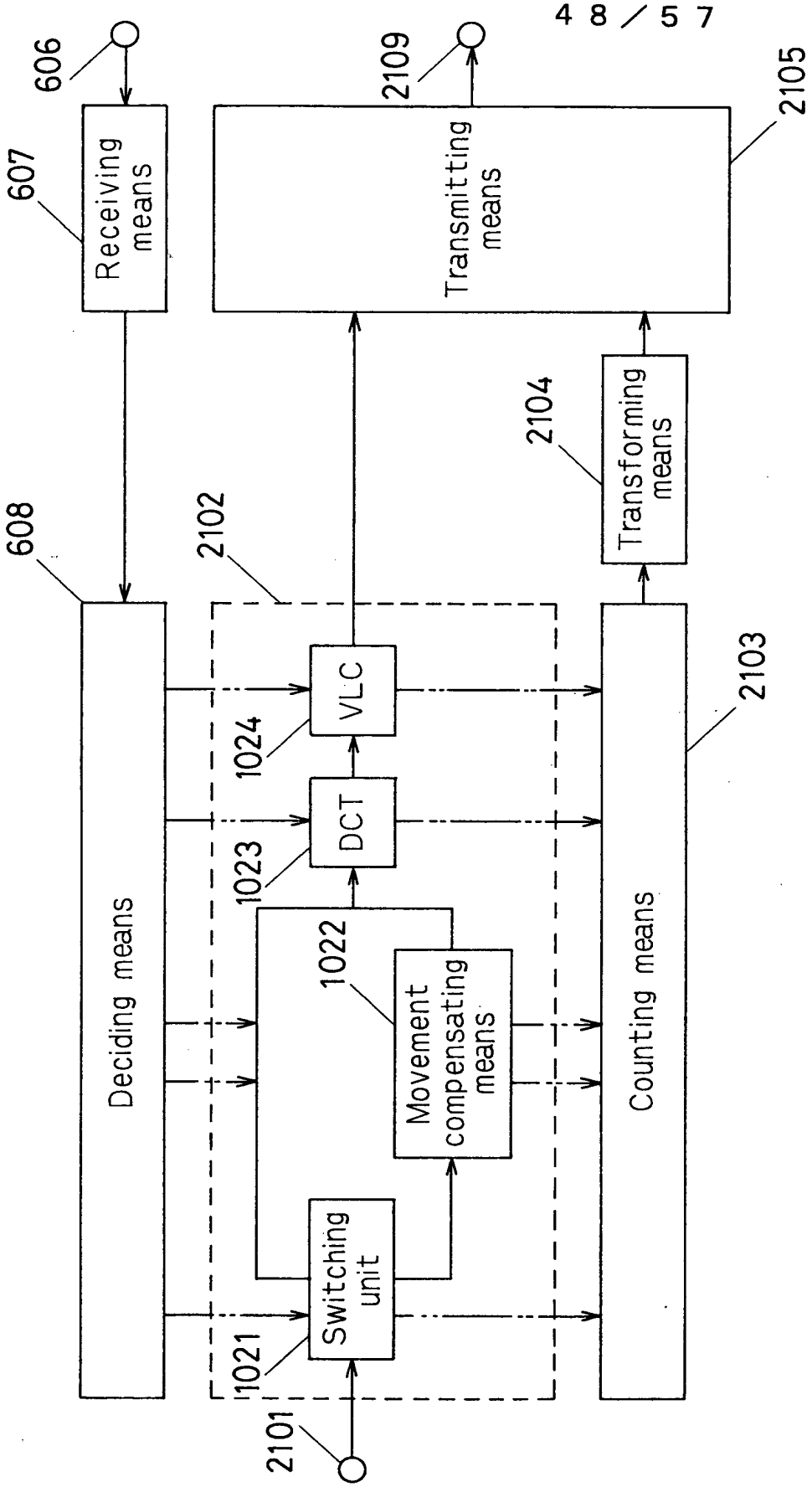

606
Receiving means
607
608
Deciding means
2101
2102
Switching unit
1021
Movement compensating means
1022
DCT
1023
VLC
1024
2103
Counting means
2104
Transforming means
2105
Transmitting means
2109

Fig. 38

Fig. 39

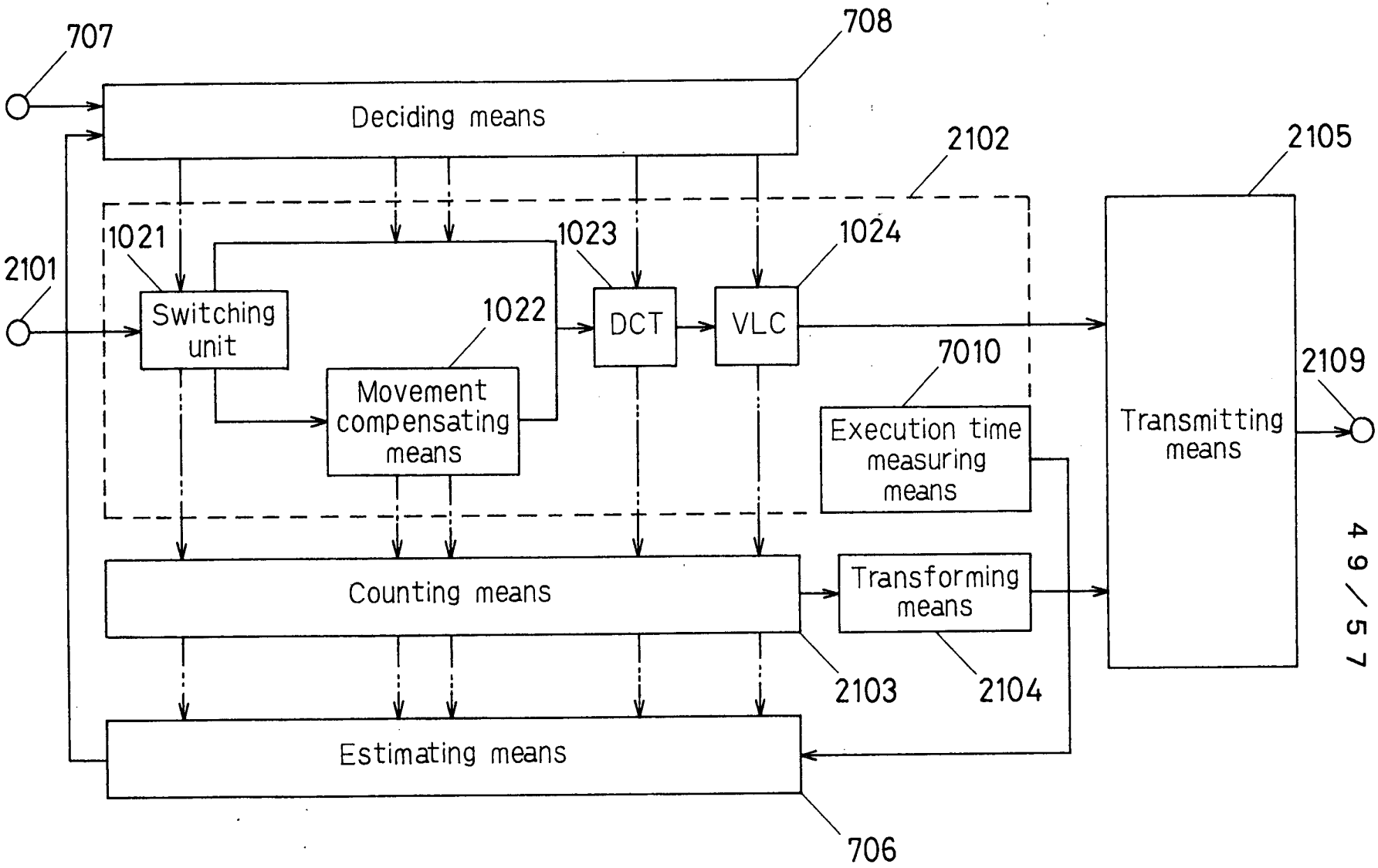

707
708
Deciding means
2102
2105
1021
1023
1024
2101
Switching unit
1022
DCT
VLC
7010
2109
Movement compensating means
Execution time measuring means
Transmitting means
Counting means
Transforming means
2103
2104
Estimating means
706

Fig. 40

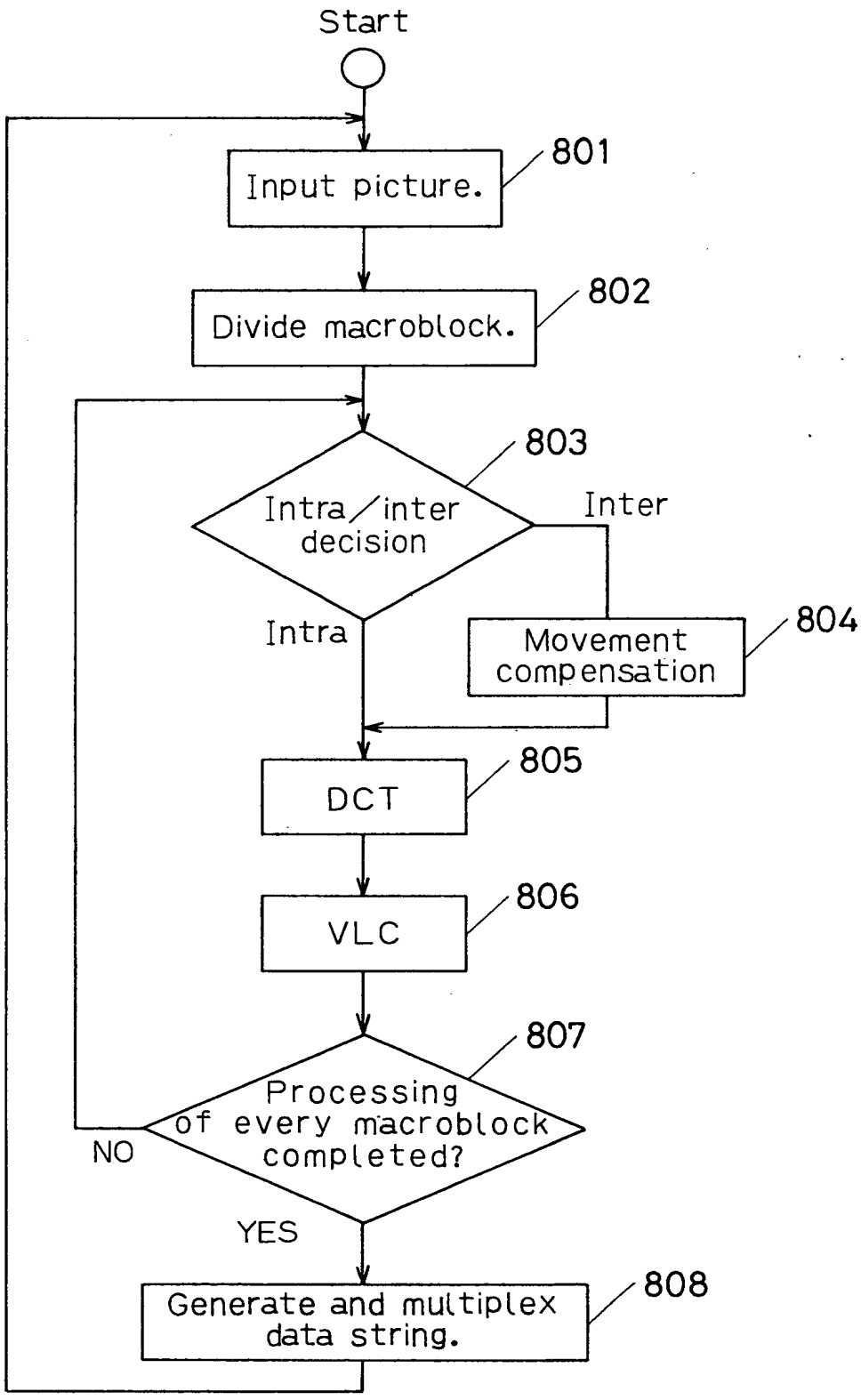

Start
Input picture.
801
Divide macroblock.
802
803
Intra/inter decision
Inter
Intra
Movement compensation
804
DCT
805
VLC
806
807
Processing of every macroblock completed?
NO
YES
Generate and multiplex data string.
808

Fig. 41

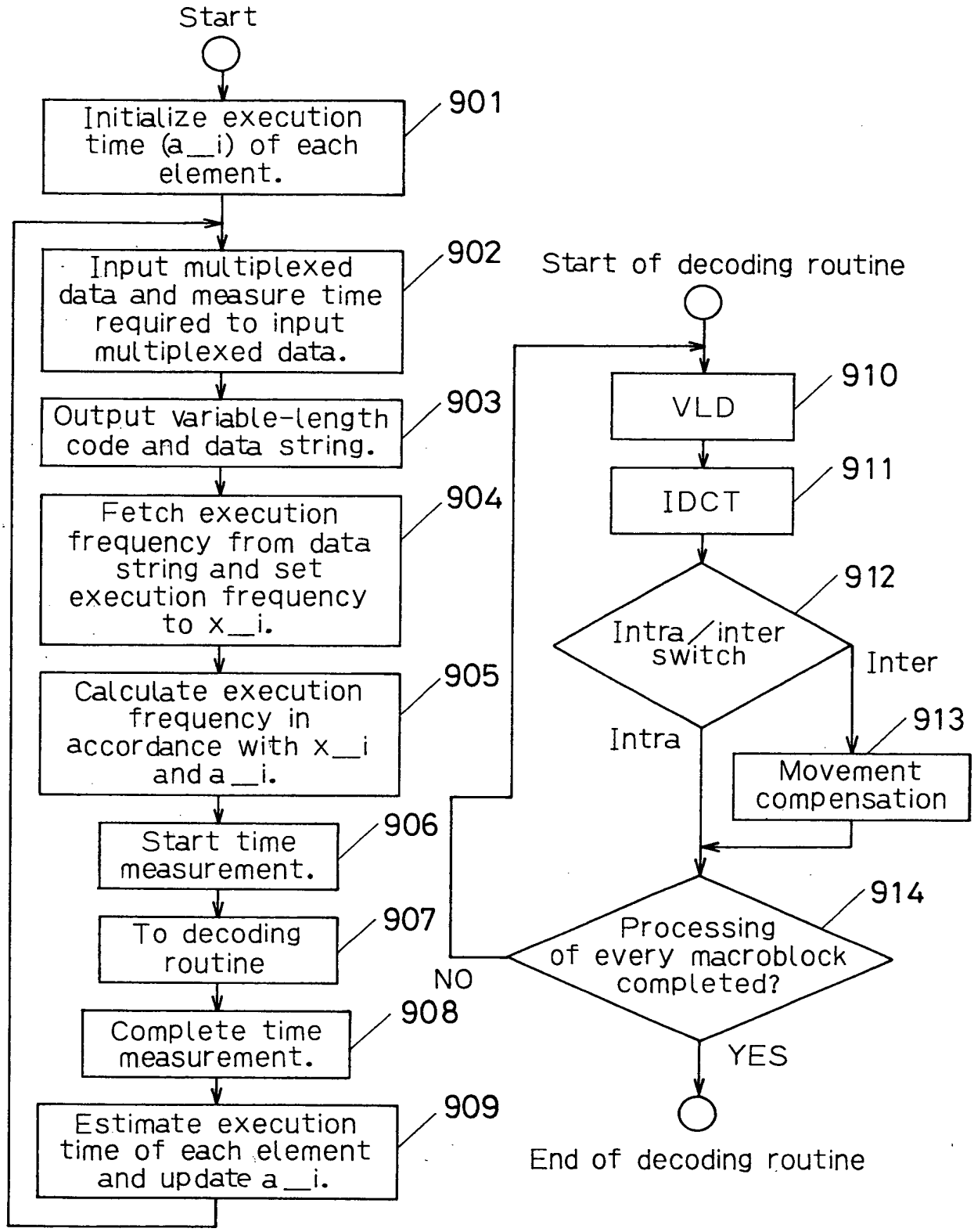

Start
Initialize execution time (a_i) of each element.
901
Input multiplexed data and measure time required to input multiplexed data.
902
Output variable-length code and data string.
903
Fetch execution frequency from data string and set execution frequency to x_i.
904
Calculate execution frequency in accordance with x_i and a_i.
905
Start time measurement.
906
To decoding routine
907
Complete time measurement.
908
Estimate execution time of each element and update a_i.
909
Start of decoding routine
VLD
910
IDCT
911
Intra/inter switch
912
Inter
Intra
Movement compensation
913
Processing of every macroblock completed?
914
NO
YES
End of decoding routine

Fig. 42

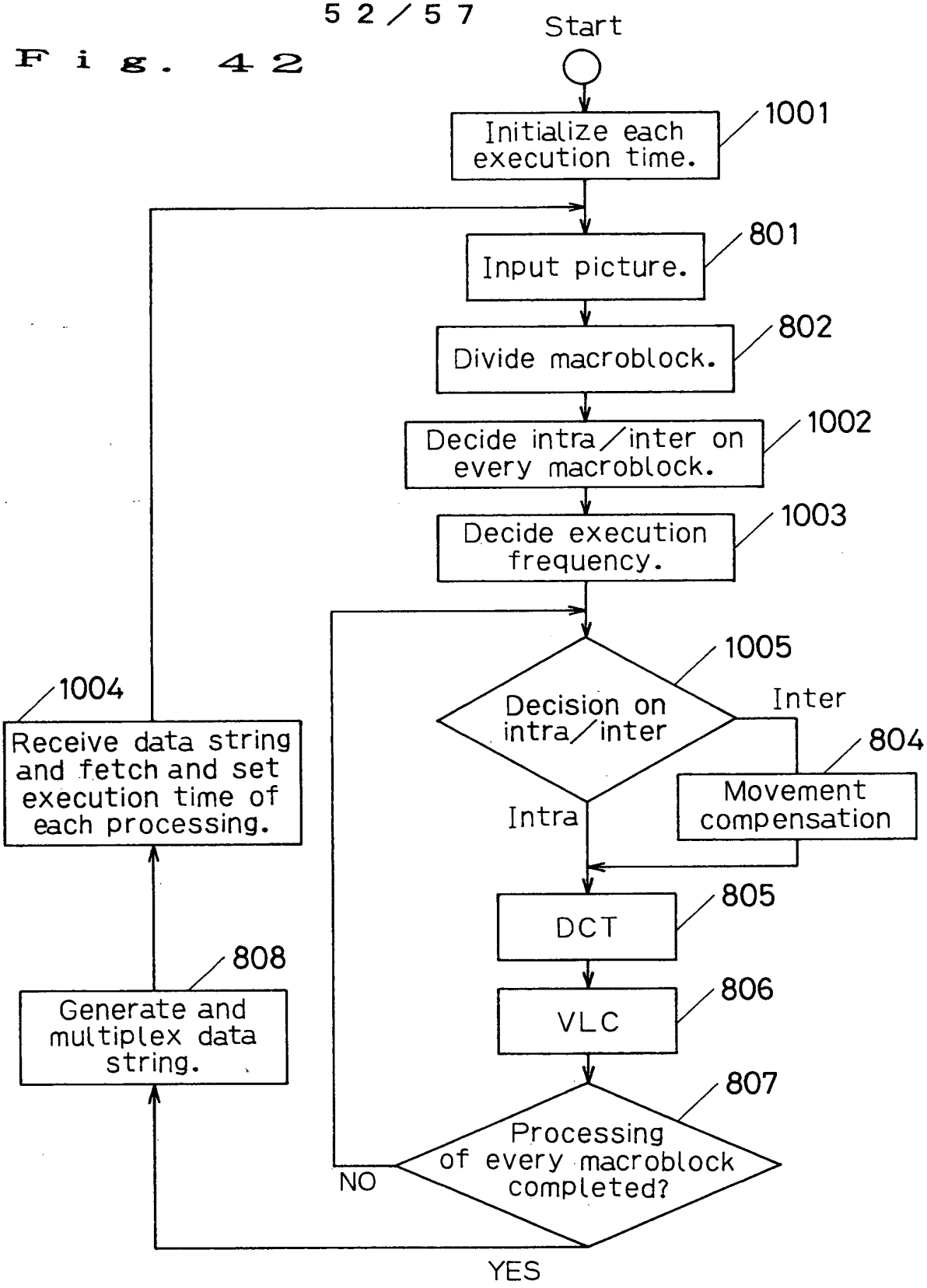

Start
Initialize each execution time.
1001
Input picture.
801
Divide macroblock.
802
Decide intra/inter on every macroblock.
1002
Decide execution frequency.
1003
Decision on intra/inter
1005
Inter
Intra
Movement compensation
804
DCT
805
VLC
806
Processing of every macroblock completed?
807
NO
YES
Generate and multiplex data string.
808
Receive data string and fetch and set execution time of each processing.
1004

Fig. 43

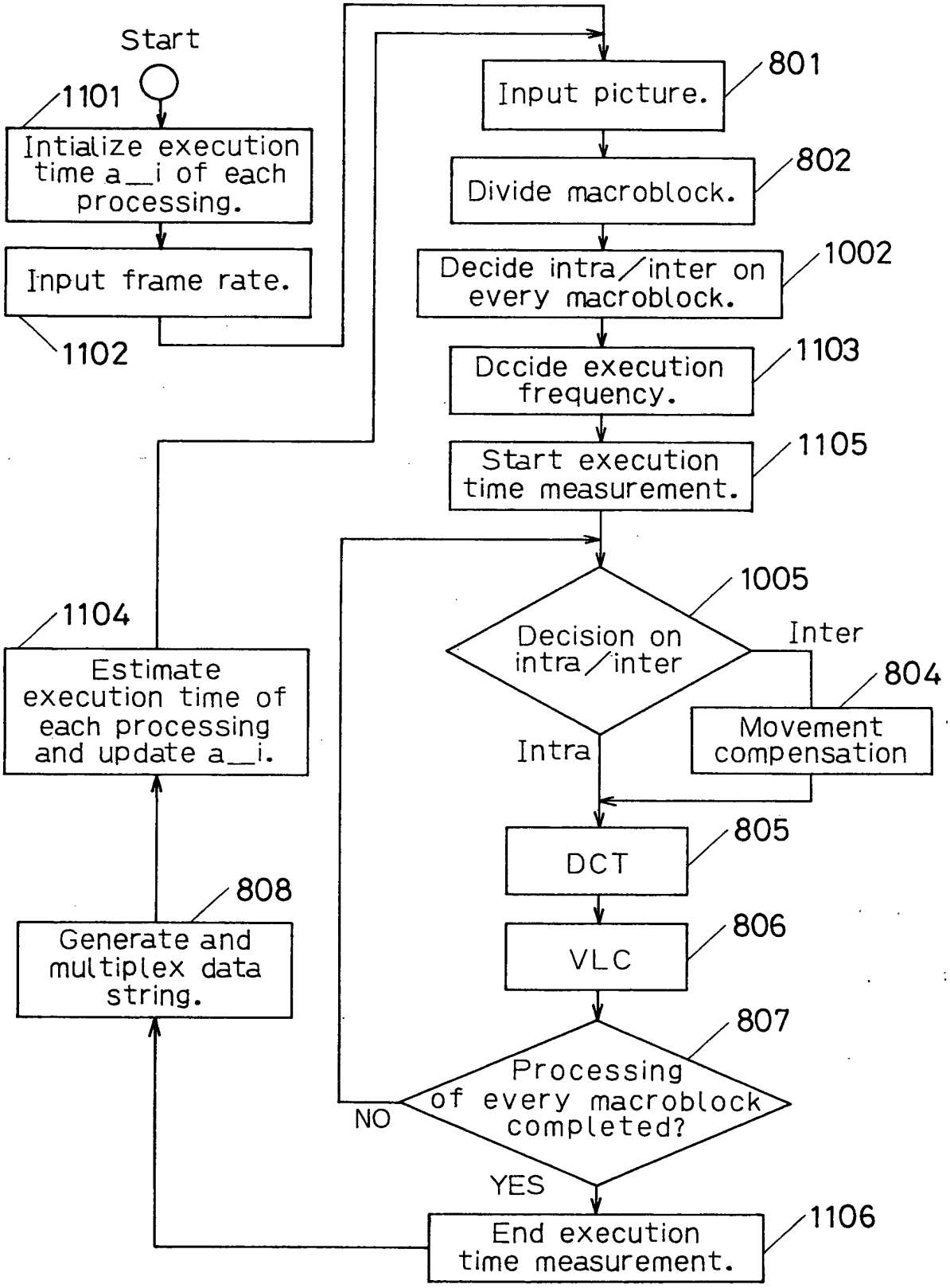

Start
1101
Intialize execution time a_i of each processing.
Input frame rate.
1102
Input picture.
801
Divide macroblock.
802
Decide intra/inter on every macroblock.
1002
Dccide execution frequency.
1103
Start execution time measurement.
1105
Decision on intra/inter
1005
Inter
Intra
Movement compensation
804
DCT
805
VLC
806
Processing of every macroblock completed?
807
NO
YES
End execution time measurement.
1106
Estimate execution time of each processing and update a_i.
1104
Generate and multiplex data string.
808

Fig. 44

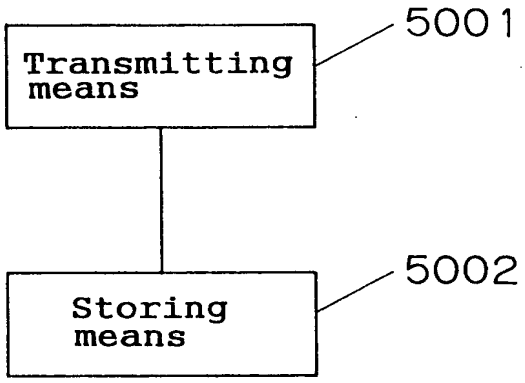
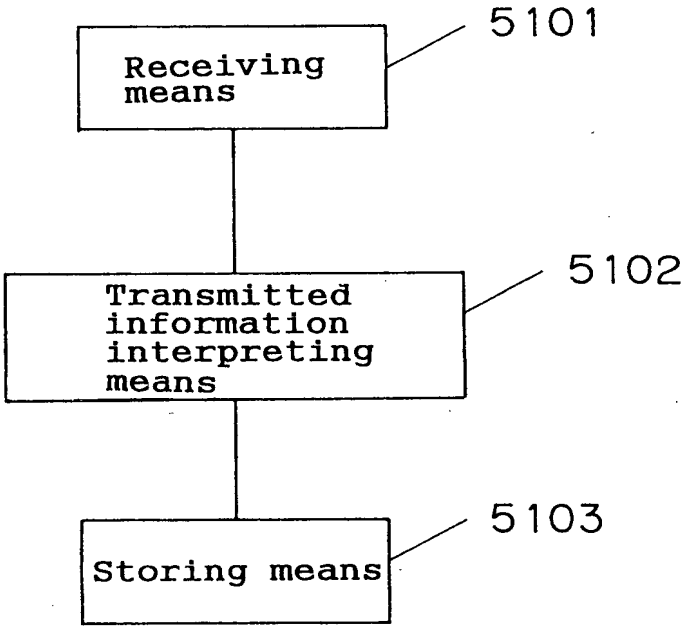


Fig. 45


Receiving means
5101
Transmitted information interpreting means
5102
Storing means
5103

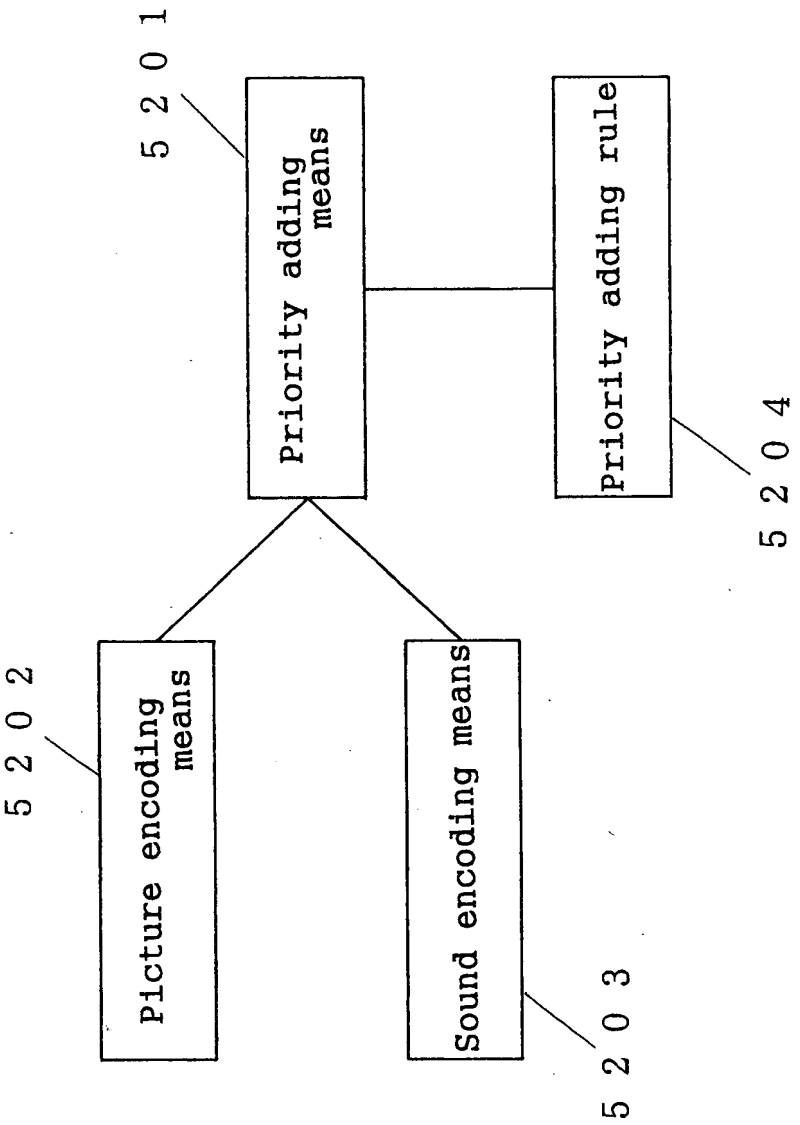

5201
Priority adding means
5204
Priority adding rule
5202
Picture encoding means
5203
Sound encoding means

Fig. 46

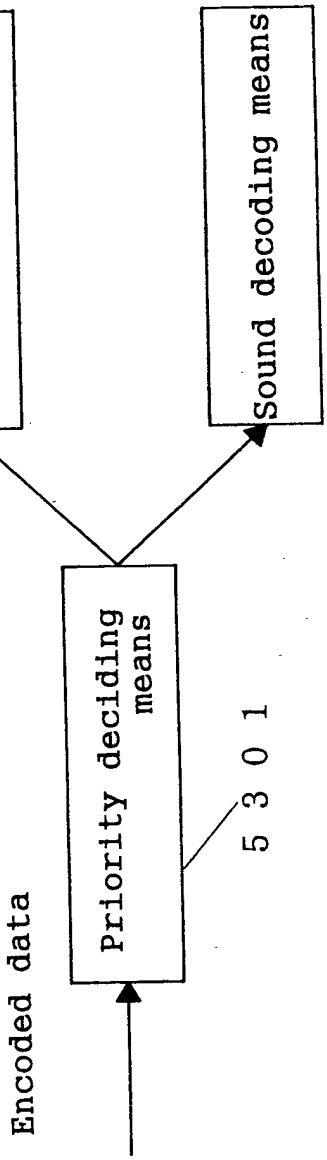

Picture decoding means
Sound decoding means
Priority deciding means
5301
Encoded data

Fig. 47